国外工业控制与智能制造丛书

HZ BOOKS 华章教育

Pearson

机械电子学

机械和电气工程中的电子控制系统

（原书第6版）

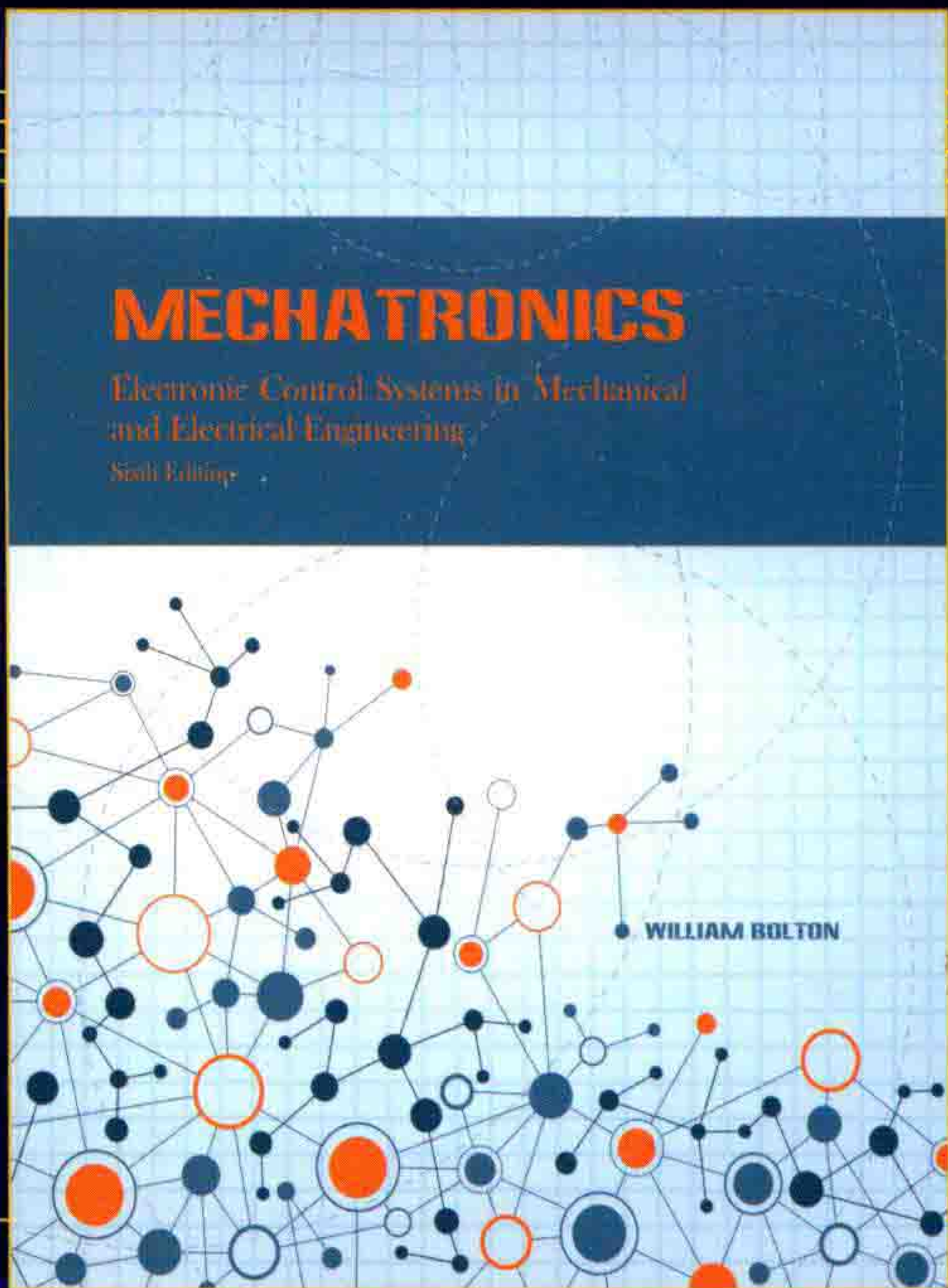
[美] 威廉·博尔顿（William Bolton） 著

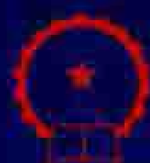
付庄 等译

Mechatronics

Electronic Control Systems in Mechanical and Electrical Engineering

Sixth Edition





机械工业出版社
China Machine Press

机械电子学是电子工程、机械工程、控制和计算机工程的集成，它处在无数设备、过程和技术的核心。从自动对焦照相机，到汽车发动机管理系统，从达到最新技术发展水平的机器人，到普通的洗衣机，都离不开机械电子学知识和技术。本书是第6版，理论讲解深入浅出，体系结构完整，机械学与电子学的知识紧密结合，习题丰富，一直被国外各大学采用，是该领域高等院校本科生、研究生或工程技术人员的必备读物。

本书对机械电子学领域做了清晰和全面的介绍，新增了更多的机电系统专题和关于机器人系统的内容，涵盖了传感器与信号调理、数字信号、数字逻辑、数据显示、气动和液压驱动系统、机械驱动系统、电气驱动系统、微处理器、输入输出系统、PLC、通信系统、故障检测、系统模型、系统的动态响应、系统传递函数、频率响应、闭环控制系统、人工智能、机电系统设计案例等，为形成一种真正的跨学科和综合化的工程方法提供了一个思维框架，是一本多学科交叉融合的综合性教材，有助于读者获得所需的综合能力来理解和设计机电一体化系统。

作者简介

威廉·博尔顿（William Bolton）曾担任英国商业与技术委员会（BTEC）继续教育的顾问以及研发和监控的负责人，他编著了多部成功的工程教科书，包括《Mechatronics》《Engineering Science》《Higher Engineering Science》《Mechanical Science》《Instrumentation and Control Systems》等，被国外各大学广泛采用，获得了很好的反响。



www.pearson.com

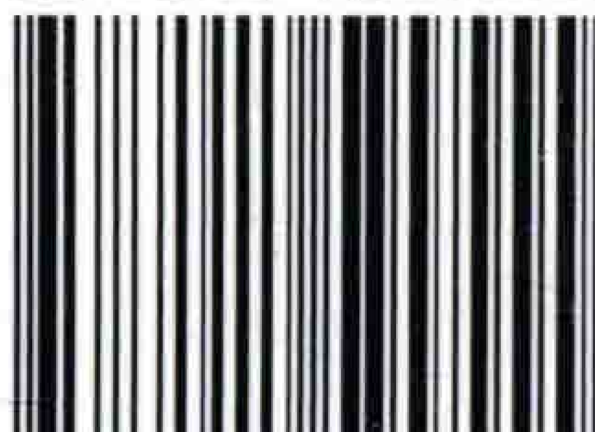
投稿热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn



上架指导：机械/电子控制

ISBN 978-7-111-59955-5



定价：129.00元

封面设计·锡彬

国外工业控制与智能制造丛书

机械电子学

机械和电气工程中的电子控制系统

（原书第6版）

[英] 威廉·博尔顿（William Bolton）著

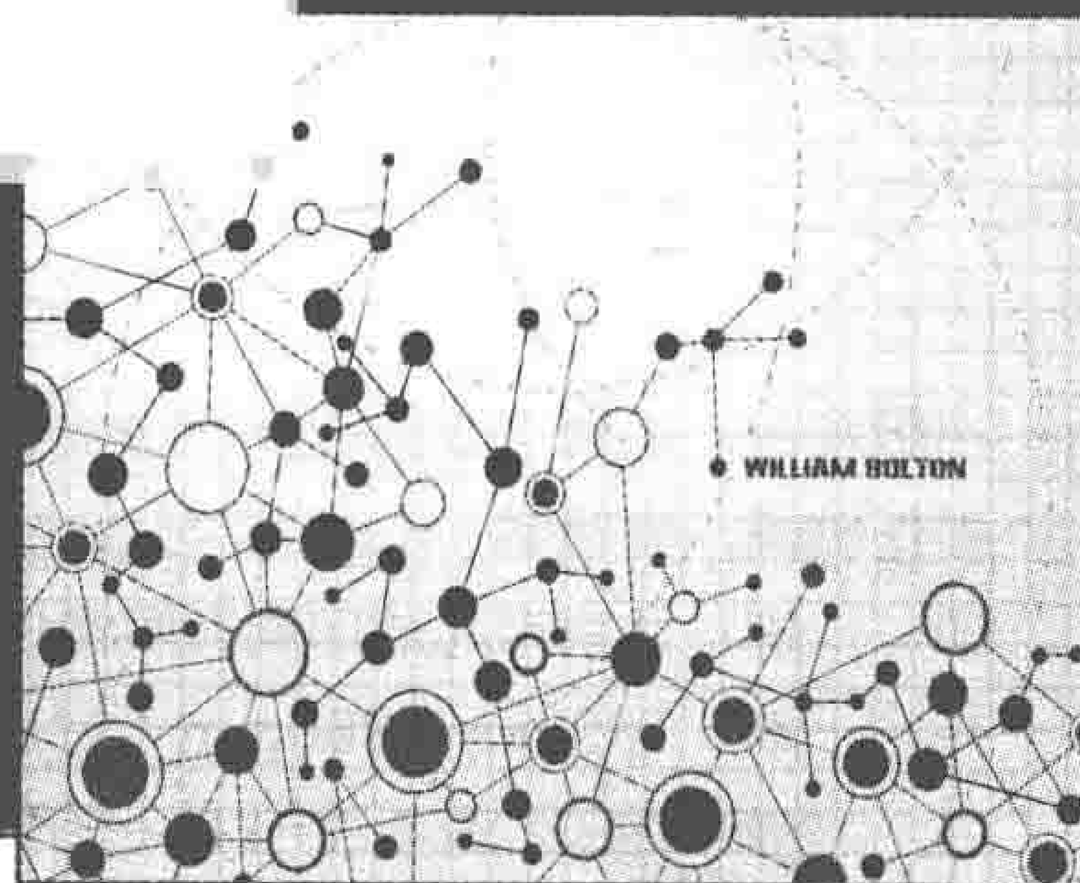
付庄 等译



Mechatronics

Electronic Control Systems in Mechanical and Electrical Engineering

Sixth Edition





机械工业出版社

China Machine Press

图书在版编目（CIP）数据

机械电子学：机械和电气工程中的电子控制系统（原书第6版）/（英）威廉·博尔顿（William Bolton）著；付庄等译．—北京：机械工业出版社，2018.5
（国外工业控制与智能制造丛书）
书名原文：Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering, Sixth Edition

ISBN 978-7-111-59955-5

I. 机…　II. ①威…　②付…　III. 机电一体化　IV. TH-39

中国版本图书馆 CIP 数据核字（2018）第102262号

本书版权登记号：图字　01-2017-6464

这是一本为工科学生写的教科书，可以作为机械电子学领域的综合性教材。本书详尽地论述了传感器与信号调理、数字信号、数字逻辑、数据显示系统、气动和液压驱动系统、机械驱动系统、电气驱动系统、微处理器、汇编语言、C语言、输入/输出系统、PLC、通信系统、故障检测、系统模型、系统的动态响应、系统传递函数、频率响应、闭环控制系统、人工智能、机电系统设计案例等方面的相关知识。

本书内容丰富，实用性强，不仅可作为高等院校机械、电气、控制工程等专业本科生和研究生的教材，也可作为广大工程技术人员的参考书。

出版发行：机械工业出版社（北京市西城区百万庄大街22号　邮政编码：100037）
责任编辑：张锡鹏　　责任校对：殷　虹
印　　刷：北京市兆成印刷有限责任公司　　版　　次：2018年7月第1版第1次印刷
开　　本：185mm×260mm　1/16　　印　　张：32.5
书　　号：ISBN 978-7-111-59955-5　　定　　价：129.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，信息学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的信息产业发展迅猛，对专业人才的需求日益迫切。这对我国教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其信息科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀教材将对我国教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson、McGraw-Hill、Elsevier、John Wiley & Sons、CRC、Springer等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Alan V. Oppenheim、Thomas L. Floyd、Charles K. Alexander、Behzad Razavi、John G. Proakis、Stephen Brown、Allan R. Hambley、Albert Malvino、Peter Wilson、H. Vincent Poor、Hassan K. Khalil、Gene F. Franklin、Rex Miller等大师名家的经典教材，以“国外电子与电气工程技术丛书”和“国外工业控制与智能制造丛书”为系列出版，供读者学习、研究及珍藏。这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也越来越多被实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着电气与电子信息学科、自动化、人工智能等建设的不断完善和教材改革的逐渐深化，教育界对国外电气与电子信息类、控制类、智能制造类等相关教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www. hzbook. com
电子邮件：hzjsj@ hzbook. com
联系电话：(010) 88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037



华章科技图书出版中心

译 者 序

本书是 William Bolton 先生编写的机械电子学国际通用教材，内容涵盖传感器与信号调理、数字信号、数字逻辑、数据显示系统、气动和液压驱动系统、机械驱动系统、电气驱动系统、微处理器、汇编语言、C 语言、输入/输出系统、PLC、通信系统、故障检测、系统模型、系统的动态响应、系统传递函数、频率响应、闭环控制系统、人工智能、机电系统设计案例等，是一本多学科交叉融合的综合性教材。该教材理论讲解深入浅出，体系结构完整，机械电子结合充分，习题丰富，一直被国外各大学采用，获得了非常好的评价。

通过翻译引进该教材，可借鉴国外机械电子教学的先进经验，有利于推动国内大学机械电子课程教学与国外大学的接轨，进而推动我国本科生和研究生课程的国际化建设。

本书不仅可以作为高等院校的机械、电气、控制工程等工科专业本科生和研究生的教材，还可作为广大工程技术人员的参考书。

本书第 3 ~ 5、9 ~ 12、14、16 ~ 19、21、23、24 章和索引由付庄翻译，第 1 章由付庄、管恩广翻译，第 2 章由付庄、蔡雄风翻译，第 6 章由宋阳翻译，第 7 章由郑望望翻译，第 8 章由邓文昊翻译，第 13 章由钱欢翻译，第 15 章由翟嘉心翻译，第 20 章由汤智诚翻译，第 22 章由刘基昊翻译。附录 A 由周航飞翻译，附录 B、C 由杨淞翻译，附录 D、F 由刘文红翻译，附录 E 由薛美风翻译，附录 G 由魏振红翻译。

第 2、4、7、10、11、13、20 章和附录由付庄初校，第 3 章由魏振红初校，第 5 章由盛致愉初校，第 6 章由邓文昊初校，第 9 章由俞港初校，第 17 章由周航飞初校，第 1、16、18、21、23 章由管恩广初校，第 8、19 章由詹世涛初校，第 22 章由曹峰初校，第 12 章由李爽初校，第 24 章由郑望望初校。付庄进行了全书的总审校，编写了原书的勘误表，并修正了书中的错误和疏漏之处。本书的翻译还得到了上海交通大学“机电一体化系统设计”研究生公共基础课建设项目的支持和许多老师、同学的帮助，并参考了相关的资料，在此对这些资料的作者以及相关的老师和同学表示衷心的感谢。

由于译者水平有限，本书错译漏译等不足之处在所难免，敬请读者批评指正。

译者
2018 年 2 月

前　言

机械电子学（或称机电一体化，mechatronics）一词最早由日本工程师于 1969 年提出。机械电子学是将机械结构学和电子学结合而成的一门新学科。到了今天，这个词有了更为广泛的含义，一般可以认为它是一种新的解决工程技术问题的思想。这种思想具体表现为在产品的设计制造和加工过程中，对机械工程、电子技术及计算机智能控制的有效并行集成。通过应用这种思想，很多原本由机械结构实现具体功能的产品都可以由包含微控制器的产品替代。而这种可替代性将使生产过程更具灵活性，重复设计和重复编程更为简单，进而可以让系统具有自动化数据收集及报告的能力。

机械电子学理论要求工程技术人员在解决工程问题时要有运用多学科、综合方法的能力。因此，工程技术人员不能仅限于掌握一个领域的技术知识，还需要有能力运用多学科知识并将它们联系起来。这就需要更多的特殊技巧。本书在介绍机械电子学的基本背景之外，还给出了上文所提到的特殊技巧的关联方法。

为了满足国家高等教育对技术人员的毕业认证要求，本书的第 1 版囊括了英国商业与技术委员会（BTEC）关于机械电子学的单元。同时本书的编写还包含介绍了许多机械电子学方面的特殊要求，如应用机械电子学方法解决问题时，需要注意的设计、生产和维护方面的问题。本书在英国及美国被广泛应用于本科生和研究生教学。结合英国和美国用书教师的反馈信息，本书第 2 版对内容进行了扩充和加深，使得本书不但对原有的读者具有参考意义，而且更适合作为本科生教材使用。本书第 3 版简化了其中的部分解释内容，对微控制器及其编程给出了更详细的讨论，增加了对机电一体化系统模型的介绍，并且就关键内容在附录中给出了索引。第 4 版对本书的内容及文字表达形式进行了全面的调整，包括重组了部分章节；将更多材料添加到附录中，以避免扰乱行文的流畅性；引入了诸如人工智能等新内容；给出了更多的实例分析；简化了不必要的章节。同时，在每一章添加了学习目的及关键点目录。第 5 版基本保持了原有的结构，只是通过与本书采用者的交流，对书中的一些内容进行了相应的增删。

由于一些教师感觉各章的排序不符合一般的教学顺序，第 6 版进行了重组。因此，新版本把系统模型部分移到微处理器系统之后。其他变化包括添加了关于 Arduino 的资料，还在第 24 章中添加了更多的题材。

本书的主要目的是帮助技术人员和在校生全面地理解机械电子学内容。本书可以在以下几方面帮助读者：

- 掌握理解和设计机电一体化系统所必需的机械工程、电子学和计算机科学方面的知识技巧。
- 掌握机械电子学中多学科知识的交叉应用。
- 学习设计机电一体化系统。

本书的每一章都包含：本章目的、本章小结、详尽的实例说明及习题。习题的答案可以在本书的结尾部分找到。第 24 章中的系统研究和设计作业的提示也在本书的结尾给出。

本书的结构为：

- 第 1 章是关于机械电子学的通识介绍。
- 第 2 ~6 章主要介绍传感器方面的知识及信号调理。
- 第 7 ~9 章介绍驱动系统。
- 第 10 ~16 章介绍微处理器/微控制器系统。
- 第 17 ~23 章介绍系统模型方面的内容。
- 第 24 章总结设计机电一体化系统时需要注意的问题。

授课者的教师指南、测验材料和 Powerpoint 幻灯片可从如下网站下载：www. pearsoned. co. uk/bolton [⊖]。

感谢书中涉及仪器的生产制造商提供的出版方面的支持。还要感谢那些仔细阅读了第 5 版并提出了改进建议的审稿人。

William Bolton

⊖ 关于本书教辅资料，只有使用本书作为教材的教师才可以申请，需要的教师请联系机械工业出版社华章公司，电话：13601156823，邮箱：wangguang@ hzbook. com。——编辑注

目　录

第三部分 驱动

第四部分 微处理器系统

第五部分　系统模型

第六部分 总结

第七部分 附录

第一部分 概 述

第1章 机械电子学导论

本章目的

通过本章的学习，读者将能够：

- 解释什么是机械电子学，以及领会它与工程设计的关系。
- 解释什么是系统，以及区分测量系统组成部分的意义。
- 描述开环系统和闭环系统中的各种表达形式及组成部分。
- 分析系统模型的需求，预测系统行为。

1.1 什么是机械电子学

机械电子学（或称机电一体化，mechatronics）一词最早由日本工程师于1969年提出。机械电子学是将机械结构学和电子学结合而成的一门新学科。到了今天这个词有了更为广泛的含义，一般可以认为它是一种新的解决工程技术问题的思想。这种思想具体表现为在产品的设计制造和加工过程中，将机械工程、电子技术及计算机智能控制有效并行集成。通过应用这种思想，很多原本由机械结构实现具体功能的产品都可以由包含微控制器的产品替代。而这种可替代性将使得生产过程具有更好的灵活性，重复设计和重复编程更为简单，进而可以让系统具有自动数据收集和报告的能力。

机电一体化系统并不是简单地将电子系统和机械系统结合而得到的。而且机电一体化系统比简单的控制系统复杂很多。机电一体化系统在设计过程中遵循统一的设计方法。这种方法将机械、电子和控制系统完整地集成在一起。这种集成化的、多学科交叉的方法被广泛应用于各种工程设计领域，诸如汽车设计、机器人制造、机械工具加工、清洗设备设计、摄像设备设计等。如果需要设计更加廉价、更加可靠、灵活性更高的系统，那么设计在初始阶段就需要对机械工程、电气工程、电子学、控制方面的内容进行集成。在处理多学科交叉集成问题时，机械电子学使用统筹方法，而不是将各个部分分开按照顺序的方式来设计，例如，先设计机械系统再设计电气部分及微处理器部分。因此，机械电子学是一种集成化的工程设计方法，具有独特的设计理念。

机械电子学将多个技术领域融合在一起。如图1-1所示，这些领域包括传感器、测量系统、驱动和执行系统、微处理器系统，以及与这些系统相关的控制及行为分析。这一点就是对本书内容本质上的一个概括。本章是机械电子学的引论，同时介绍一些基本概念，并给出其他章节的组织框架。具体的技术内容细节将在后面的章节逐一介绍。

1～3

1.1.1 机电一体化系统实例

我们以现代的自动对焦、自动曝光照相机为例。在使用这种照相机的时候，你需要做的只是将它对准要拍摄的物体，然后按下按钮，就可以完成照相了。照相机本身可以自动调节焦距使被照物体位于焦距上，还可以自动调节光圈和快门速度以取得正确的曝光。使用者不需要手

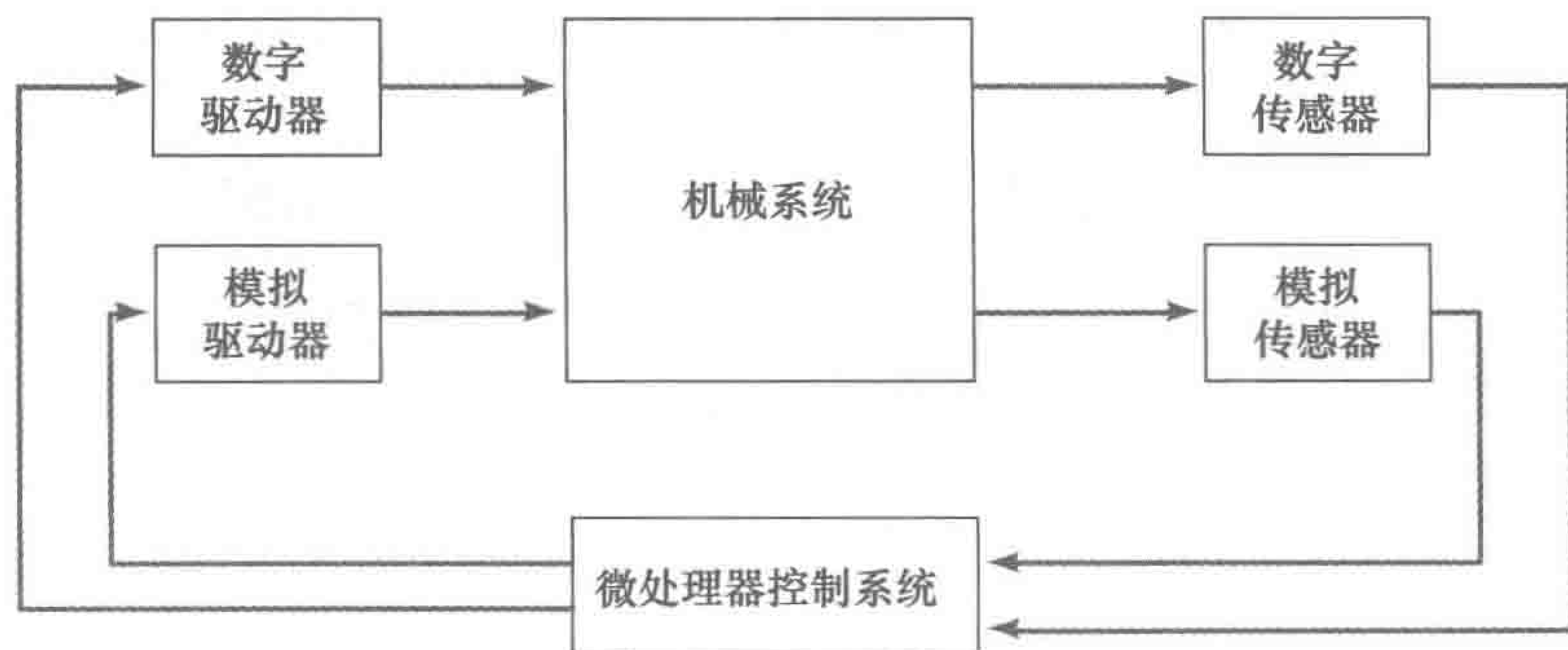


图 1-1　机电一体化系统的基本要素

动调节焦距、光圈和快门速度。再以卡车上使用的智能悬挂系统为例。这种悬挂系统可以在搭载不均匀负载的时候保持车身平衡；可以在转弯、路况较差的情况下，保证驾驶平稳。再以自动化生产线为例。这样的一条生产线可以包含很多的生产过程。这些生产过程按照正确的顺序自动执行，同时这些过程的每一个输出状态都会自动输出到需要的地方。可以说，自动照相机、卡车的悬挂系统和自动化生产线都是机械工程、电气工程及控制系统相互结合的完美实例。

1.1.2　嵌入式系统

嵌入式系统是通过将微处理器嵌入到设计中而形成的系统。在机械电子学领域，我们广泛地关注这种系统。微处理器在本质上可以被认为是逻辑门和存储单元的集合。但是，这些逻辑门和存储单元并不是分别作为个体由导线相互连接。微控制器主要通过软件的编写来实现不同的逻辑功能。作为一个逻辑门表示意义的说明，我们可能需要在输入 A 和输入 B 都有信号的时候得到一个输出信号。这可以通过一个“与”逻辑门来实现。而一个“或”逻辑门则可以保证只要输入 A 和输入 B 有一个通道上有信号，就可以输出一个信号。因此，微处理器的主要功能是观察输入通道的开关状态，依据内部的程序处理观察到的结果，并控制输出通道的开关。关于微处理器的更多细节将在第 10 章介绍。

对于应用于控制系统的微处理器，它需要额外的芯片来提供存储数据的空间及与外界交换信号的接口。**微控制器**就是一种将微处理器功能及上述额外芯片功能集成在一起的微处理器。

嵌入式系统是基于微处理器的系统。这种系统具备一系列的功能，但是这些功能不能像计算机一样可以由最终的使用者通过编程来改变。因此，在使用嵌入式系统的时候，使用者无法通过增加或者替换软件的方式来改变系统的功能。

4

举一个在控制系统中使用微控制器的例子。例如，在现代洗衣机中安装了微控制器控制系统，于是这台洗衣机就可以自动控制洗涤循环、给排水、电动机和水温。在现代汽车工业中，微控制系统主要用来控制防抱死系统及发动机管理系统。除此之外，还有很多其他应用嵌入式系统的例子，包括自动对焦、自动曝光照相机、便携式摄像机、手机、DVD 播放器、电子读卡器、复印机、打印机、扫描仪、电视机和空调。

1.2　设计流程

任何系统的设计流程都可以分为以下几个阶段。

1. 需求

设计过程的开始就是分析消费者或者客户的需求。一般情况下，我们通过对潜在客户的市场调查来预测需求。

2. 问题分析

设计的第一个阶段是找出问题的本质，也就是分析问题。这是一个十分重要的阶段。因为如果问题没有准确地定义，那么设计将变得非常耗时而且得出的设计通常难以满足需求。

3. 创建规格说明文档

在对问题进行分析后就可以开始准备规格说明文档了。准备工作包括陈述问题、介绍选择解决方法时要考虑的约束条件以及判定设计质量的标准。在陈述问题的时候，所有在设计中需要实现的功能及系统特性都要准确地描述出来。因此，规格说明文档一般包含质量、尺寸、所需动作的类型及范围、精度、输入/输出部分的要求、接口、电源要求、工作环境、相关标准及测试代码等。

4. 形成可能的解决方案

这一阶段通常称为**概念性设计阶段**。为了得到问题解决方法的大体轮廓，我们分析系统必须具备的功能并尽量多地给出实现这些功能所必需的细节条件，诸如近似尺寸、形状、材料和花费。同时，我们也要找出之前类似问题的解决方法。什么事情都从零做起是不必要的。

5. 选择合适的解决方案

我们评估所有可能的解决方案，并选出最优的解决方案。评估通常是指对系统进行重新建模，然后通过仿真预测系统对输入的响应。

6. 完成细节设计

在这一阶段，将完成所有选定方案的细节设计。为了实现设计细节方面的最优，有时候需要生产一些原型机或者实物模型。

7. 工程绘图

在设计选定后，具体的设计就要以工程图、电气连接图等方式表现出来。这样就完成了一个设计流程。

5

设计过程的各个阶段并不是简单地按照先后顺序一个一个执行。设计者经常需要回到前面的阶段进行更深入的思考。当我们处在形成可能的解决方案的阶段时，就经常需要回到之前的问题分析阶段进行重新思考。

1.2.1　传统的机械电子学设计

工程设计是一个复杂的过程，需要交叉使用多种技巧及多学科的知识。在传统的设计过程中，机械部分由机械工程师设计，控制系统由控制系统工程师设计。这称为顺序设计方式。然而，机械电子学的基本方法是在设计过程中将机械工程、电子学、计算机技术及控制工程这些学科以统筹的方法整合在一起考虑。机械电子学方法的内在统筹性主要依靠系统建模、模型系统对输入响应的仿真和真实系统对真实输入的响应结果分析。

为了说明如何将多学科的方法应用于问题的解决方案设计，我们以磅秤的设计作为一个分析实例。设计磅秤一般只需要考虑如下几个过程：弹簧的压缩形变，一种可以将这种形变转化为轴的转动的机械结构，轴带动指针在刻度盘上的移动。还有一个需要考虑的问题是，磅秤的测量结果不能因被测试者站在磅秤的不同位置而发生改变。但是，如果我们以超出纯粹的机械设计领域的角度来看这个问题，那么我们还可以考虑其他的一些可能性。比如，我们可以用应力测试仪替换原有的弹簧，还可以通过使用微处理器将测试仪输出的重量读数显示在 LED 电子屏幕上。这样设计出来的磅秤在结构上更为简单，减少了运动部件的使用。原有由机械结构实现的复杂功能都转为由软件来实现。

下面再举一个更复杂的实例。传统的家用中央加热系统的温控设计使用的是一种闭环双金属温度控制系统。由于双金属条的弯曲度会随着温度的变化而发生改变，所以可以用于控制加

热系统的开关。而一个多学科融合的解决方案可以通过使用带热传感器的微控制器控制系统来实现温控功能。这样的系统比传统的双金属温度控制系统具有更多优点。双金属温度调节器控制精度比较差，无法准确地控制温度。同时我们很难设计一种控制方法能够在一天的不同时间保证将温度控制在不同的值。但是微处理器控制系统就可以很容易地实现精确的程序化的温度控制。而且这样的系统也更加灵活。相比于传统的控制系统，机电一体化系统普遍具有更好的灵活性。

1.3 系统

在机电一体化系统的设计过程中，步骤之一是进行系统建模。系统模型主要用于预测输入响应发生时系统的反应动作。在建立模型时，我们通过绘制框图的方法来描述系统。**系统**是一个有输入/输出的盒子或框图，我们关心的并不是盒子内发生了什么，而是输出和输入之间的关系。当我们用数学公式描述一个真实系统的行为时，就要用到**建模**（modelling）这个术语了。而这些数学公式表达了系统的输入和输出之间的关系。例如，弹簧可以视为一个以力 F 作为输入并且以伸长量 x 作为输出的系统（见图 1-2a）。用于建模输入与输出之间关系的公式为 $F=kx$，其中 k 是常量。再比如，电动机可以视为一个以电功率为输入并且以电动机轴的转动为输出的系统（见图 1-2b）。

测量系统是一个可用于测量的盒子。它有被测量的输入量和对应的输出值。例如，一个温度测量系统，即温度计，就有温度输入和某一标准尺度下的数字输出（见图 1-2c）。

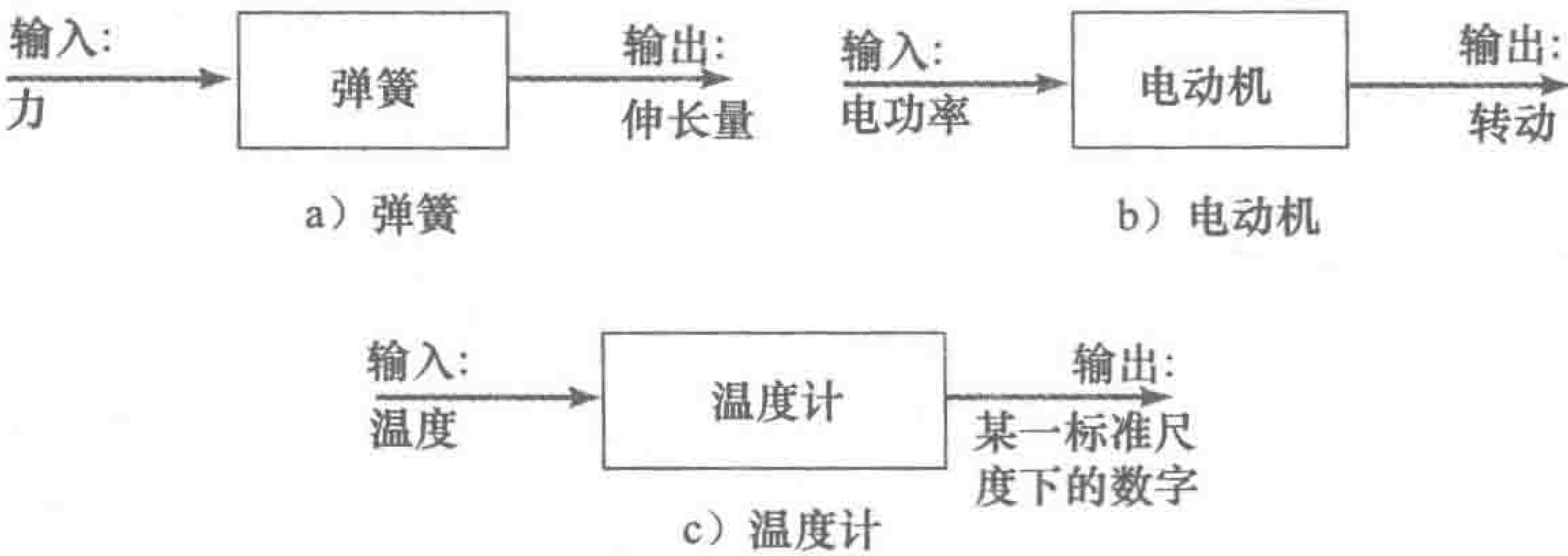


图 1-2　系统的例子

1.3.1 系统建模

任何系统对输入的响应都不是瞬时完成的。例如，对于图 1-2a 中描述的弹簧系统，尽管输入（力 F）和输出（伸长量 x）之间的关系可由 $F=kx$ 表示，但这只描述了稳态条件满足时的关系。当弹簧加上力时，在弹簧达到稳态的伸长值前，弹簧将可能会有振荡发生（见图 1-3）。系统的响应是时间的函数。因此，为了知道一个系统对输入的表现情况，我们需要建立输出对输入的模型，以便给定一个输入，我们能计算输出是如何随着时间变化的，以及输出的稳定值是什么。

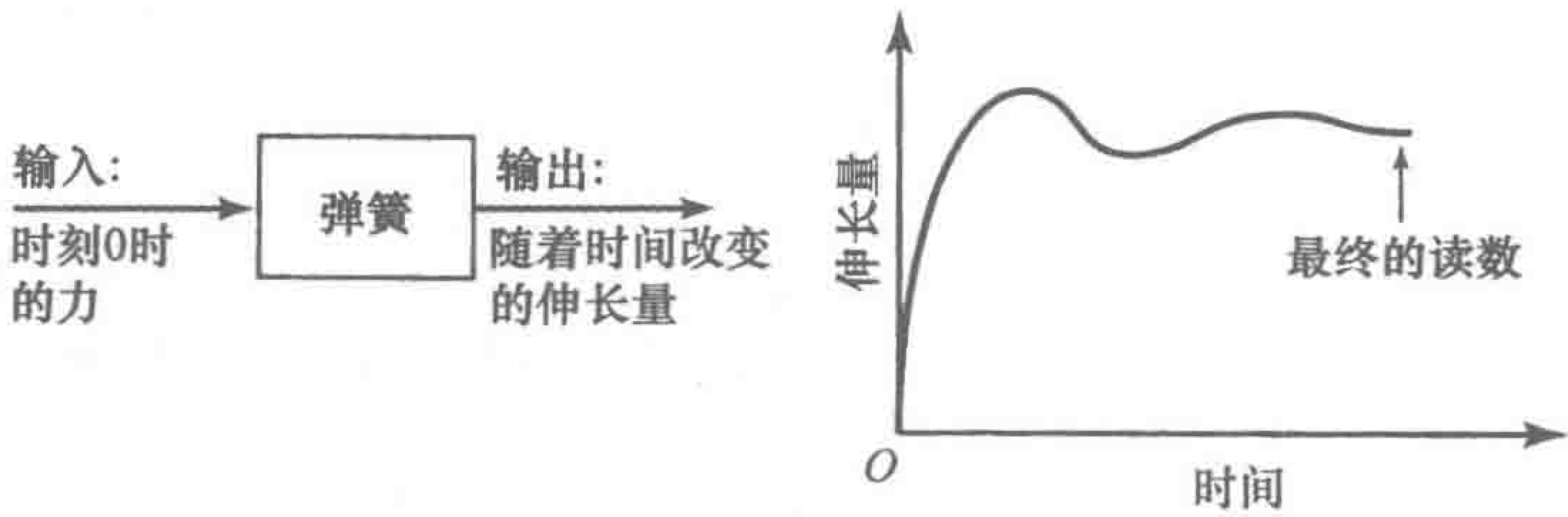


图 1-3　弹簧的输入响应

再比如，如果你打开一个电水壶的开关，水壶里的水需经过一段时间才达到沸点（见图 1-4）。同样，当微处理器的控制器给出一个信号来移动镜头进行相机自动对焦时，在镜头达到正确的焦点前，也将需要一定的时间。系统的输入和输出之间的关系常常由不同的方程来描述。这些方程和系统将在第 17 章中进行描述。

7

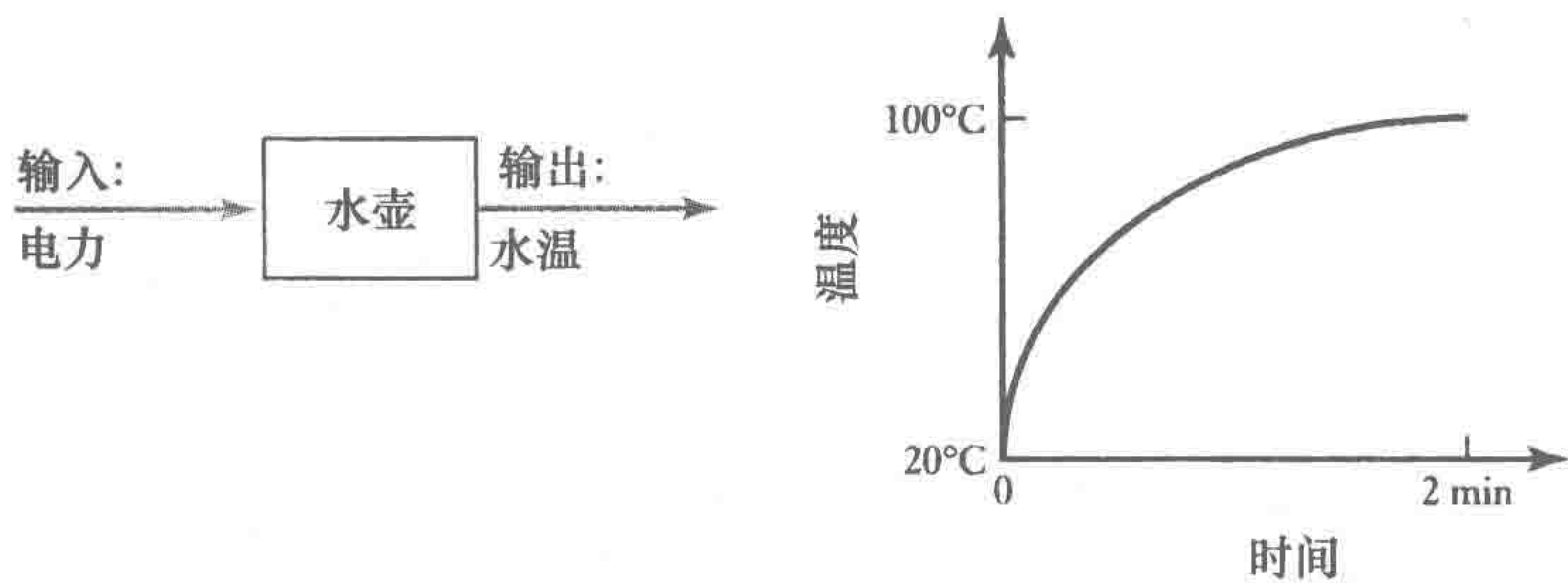


图 1-4　水壶系统的输入响应

1.3.2　连接系统

除了最简单的系统外，把系统视为一系列相互连接的方框通常是很有用的。每个这样的方框都有特殊的功能。于是，我们可以把系统的一个方框的输出变为另一个方框的输入。用这种方法绘制系统，必须认识到方框两侧的线条用箭头指示了信息流的方向，而并非物理连接。CD 播放器就是一个这样的系统连接例子。我们可以认为其有三个相互连接的方框：有 CD 输入和电信号输出的光盘驱动器；有这些电信号输入和更强电信号输出的放大器；有电信号输入和声音输出的喇叭（见图 1-5）。关于另外一组连接方块的实例将在 1.4 节中介绍。

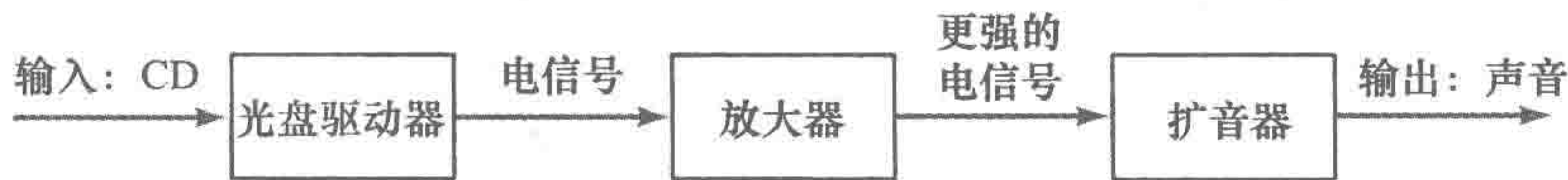


图 1-5　CD 播放器

1.4　测量系统

测量系统是机械电子学特别重要的系统之一。通常，它由三个基本元素组成（如图 1-6 所示）。

1）**传感器**：传感器可对被测的物理量做出反应，并输出与这些量相关的电信号。例如，热电偶是一种温度传感器。它的输入是温度，输出是与温度值相关的电动势（e. m. f）。

2）**信号调理器**：信号调理器获取传感器的输出信号，并将其转变成适于显示或执行控制的状态。例如，热电偶的输出是相当小的电动势信号，可被输入到放大器以获得更大的信号。这个放大器就是一种信号调理器。

8

3）**显示系统**：显示系统显示信号调理器的输出。例如，这可能是一个在刻度表上移动的指针或一个数字读数。

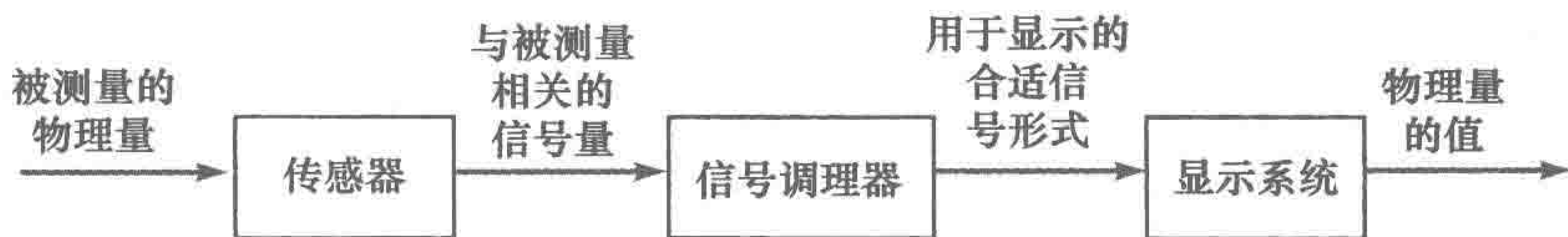


图 1-6　测量系统及其组成元素

考虑图 1-7 中数字温度计的例子。传感器的温度输入可能是半导体二极管。在恒定电流下，传感器的电势差输出是温度的度量。于是，电势差经过运算放大器的放大，变为可以直接驱动显示器的电压。传感器和运算放大器可以合并到同一个芯片上。

传感器将在第 2 章进行讨论，信号调理器将在第 3 章进行讨论，而包括所有元件的测量系统将在第 6 章进行讨论。

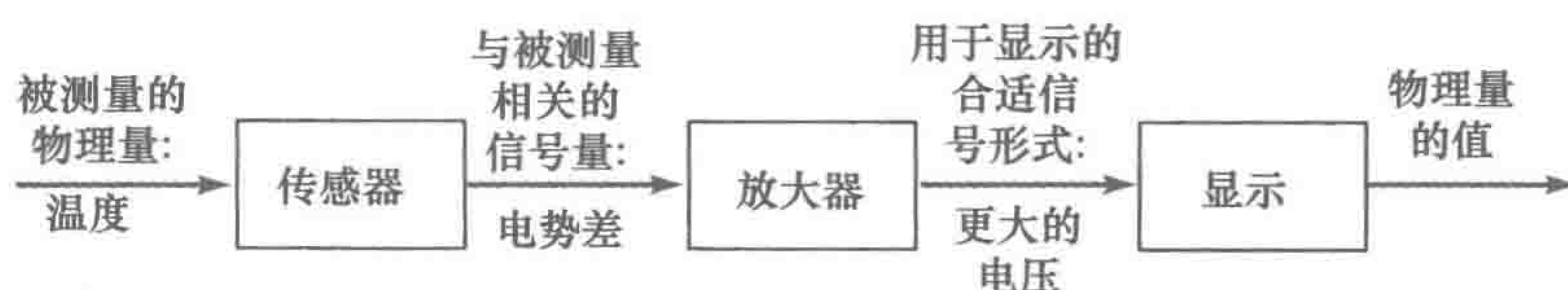


图 1-7 数字温度计的例子

1.5 控制系统

控制系统可用于：

1）控制某些变量到某些特殊的值。例如，将中央加热系统的温度控制到某一数值。

2）控制事件的序列。例如，当把洗衣机的表盘设成“白色”类型时，将洗衣机控制到特殊的洗涤周期，即事件的序列，以适于相应的洗衣类型。

3）控制一个事件是否发生。例如，只有当一个安全保护装置到位后，机器上的安全锁才能被操作。

1.5.1 反馈

举一个和我们每个人都有关的控制系统的例子。除非你病了，否则无论你是在冷还是热的温度环境中，你的身体温度几乎都保持不变。为了保持温度稳定，你的身体有个温度控制系统。如果你的体温开始上升并高于常值，你将出汗；如果温度降低，你将颤抖。这两种机制都用于恢复你的体温到正常值。这个控制系统一直保持温度的稳定。该系统有来自传感器的输入，告诉系统温度是多少，并将该数据与应该达到的温度相比较，并根据比较的结果产生合适的响应以获得需要的温度。这就是**反馈控制**的例子：信号从输出反馈回来，即得到实际温度，以便修改身体的反应，使其恢复到正常的温度值。反馈控制是通过控制系统比较系统的实际反馈输出与所需的目标值，然后相应地调整其输出来进行的。图 1-8a 说明了这一反馈控制系统。

9

控制集中供暖的房子温度的一种方法是，一个人站在有温度计的炉子的 on/off 开关旁边，根据温度计读数打开或关闭炉子。这是一个用人作为控制元素的原始反馈控制方式。为了修改输入，需要来自输出的信号，因此应用了反馈这个术语。更常见的反馈控制系统有一个恒温器或控制器，它能根据设定温度和实际温度的差值自动地打开或关闭炉子（见图 1-8b）。这一控制系统可保持温度的恒定。

如果你走过去拿起椅子上的铅笔，就需要用一个“控制系统”来保证手实际停在铅笔的上面。这是通过观察手相对于铅笔的位置，然后调整手向铅笔移动的位置来实现的。由于手的实际位置反馈信息，使你可以修改你的反应来给出需要的手的位置和运动（见图 1-8c）。这一控制系统一直控制手的定位和运动。

反馈控制系统随处可见，不仅在自然界和家里有，在工业界也广泛应用。很多的工业生产过程和机器设备需要人手动或自动控制。例如，有些过程控制的温度、液位、流速、压力等需要保持不变。这样，在化学生产过程中可能需要保持罐体里的液位或温度到某一特殊值。也有一些控制系统可以连续精确地定位某一移动零件或保持某一固定速度。例如，这可能是某一恒定速度运行的电动机，也可能是某一加工过程，可自动地控制工具位置、速度和操作。

10

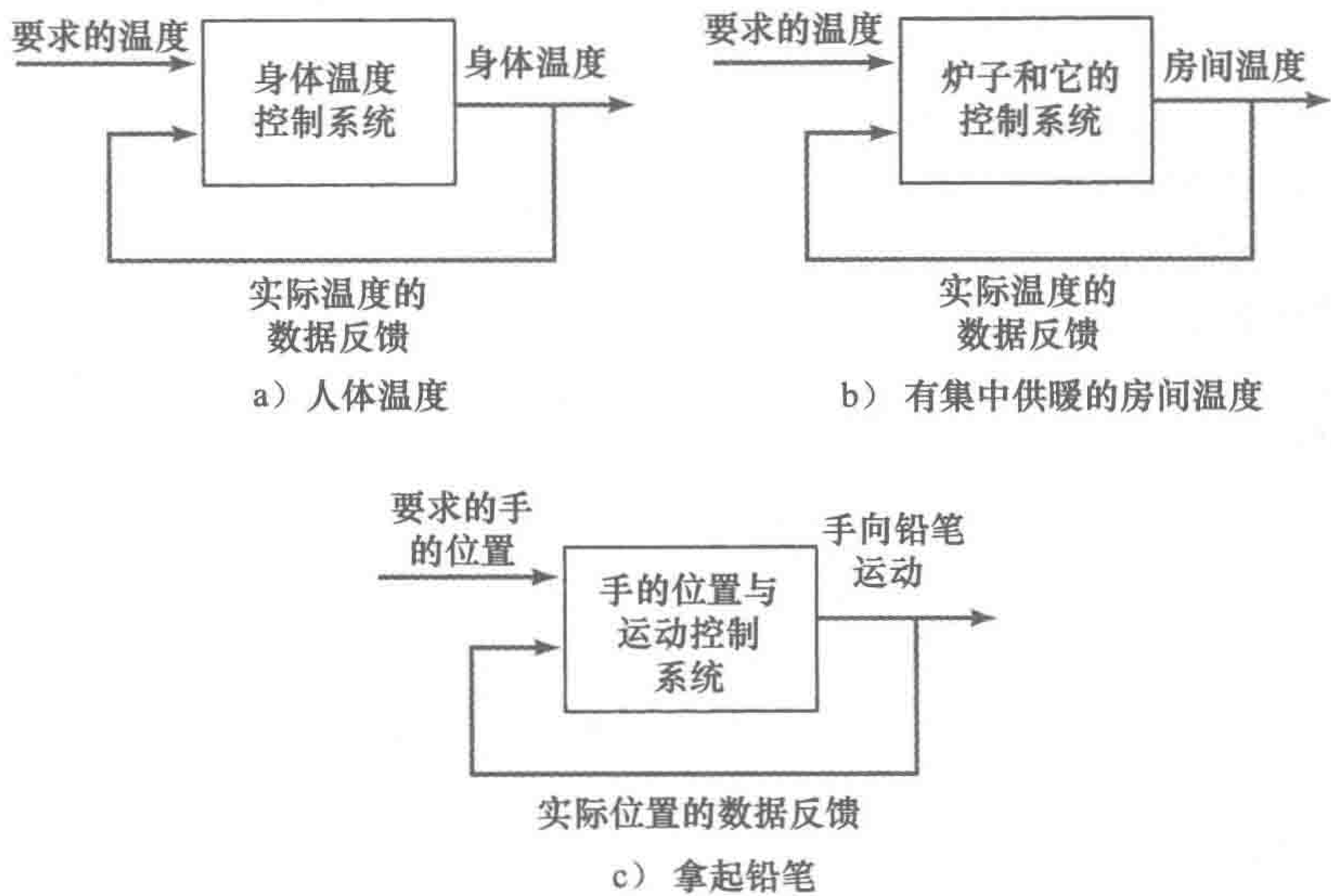


图 1-8　反馈控制

1.5.2　开环与闭环系统

控制系统的基本形式有两种，一种称为**开环**（open loop）控制系统，另一种称为**闭环**（closed loop）控制系统。二者之间的区别可通过一个简单的例子来说明。考虑一个电炉，有一个可选择 1kW 或 2kW 加热元件的选择开关。如果某个人用加热元件加热房间，并且房间不需要达到特别高的温度，他可能打开 1kW 的加热元件。房间将加热并达到某一温度，该温度只取决于打开了 1kW 而非 2kW 的加热元件这一事实。如果条件发生了改变，例如某人打开了窗户，没有热输出的相关调节途径。在这一开环控制的例子中没有任何信息反馈到控制元件上进行自动调节并保持某一恒定温度。如果这个人有温度计，并根据实际温度和期望温度之间的差打开或关闭 1kW 或 2kW 加热元件，带有加热单元的加热系统将被设计成闭环系统，以保持房间的温度恒定。这种情况下存在反馈，根据系统的输出是否是期望的温度来调节系统的输入。这意味着对开关的输入量依赖于实际温度与期望温度之间的偏差。该偏差是由比较单元决定的，在本例中比较单元是这个人。图 1-9 说明了这两种系统类型。

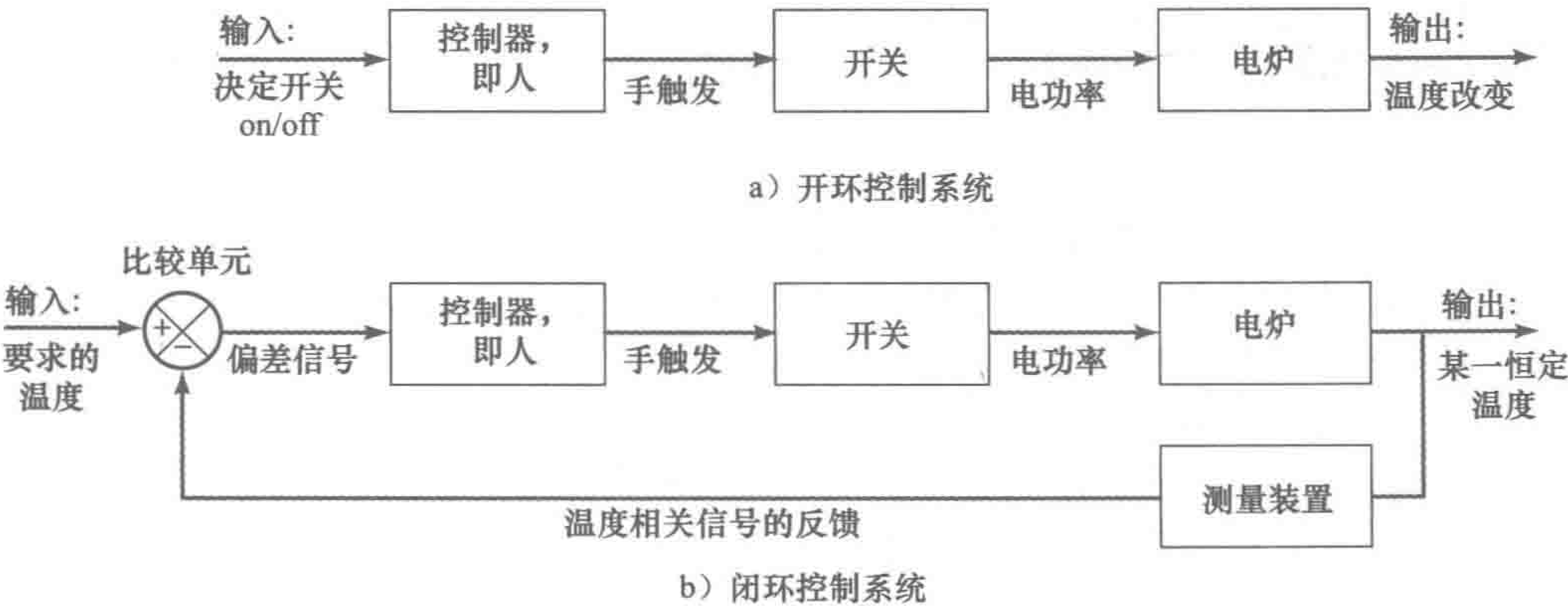


图 1-9　加热某一房间

一个日常开环控制系统的例子是家用的烤箱。它通过设定计时器来决定某一面包的烤制时间长短，从而实现烤箱的控制。烤制的褐色深浅仅仅由预设的时间决定。没有反馈来控制面包的褐变程度，以达到所需的褐色。

为了进一步说明开环与闭环系统之间的区别，我们考虑一台电动机。对于一个开环系统来说，电动机轴的旋转速度可能只取决于旋钮的初始设定，该设定影响施加到电动机上的电压。在开环系统中，任何电源电压的变化，以及由于温度的波动引起的电动机的特性变化或电动机轴的负载作用，将都会改变电动机轴的速度，却得不到补偿。这种开环控制不存在反馈回路。然而，对于一个闭环系统，控制旋钮的最初设定将对应一个特定的轴速度，这一速度将通过反馈得以保持。而且无论电源电压如何变化，电动机系统对输入信号都不会产生任何影响。在闭环控制系统中，输出信号对输入信号会产生影响，通过修改输入信号使输出信号保持到所期望的值。

开环系统的好处是相对简单，因此成本低，一般有较好的可靠性。然而，因为没有校正误差，所以它们往往是不精确的。闭环系统的好处是实际输出值可以比较准确地匹配所期望的值。然而，闭环系统更复杂、更贵，且由于部件多，导致故障的机会更大一些。

1.5.3 闭环系统的基本要素

图 1-10 给出了基本闭环系统的一般形式。它由如下几个基本要素组成。



图 1-10 一个闭环控制系统的组成要素

1. 比较单元

在被控制的可变条件下，该单元比较期望值（或参考值）与测量值，产生一个误差信号。这种情况可以视为加上一个正的参考信号和一个负的测量信号：

误差信号 = 参考信号 - 测量信号

通常，比较单元使用的符号是一个分割圆，信号输入到分割圆的各扇形区域中进行相加。由于所有的输入是加在一起的，所以反馈输入被标记为负值，而参考信号为正值，这样二者之和给出了输入信号之差。**反馈回路**（feedback loop）是一种工具，即与实际条件相关的信号被反馈回来，进而修改输入信号，再发送给某一过程。当反馈信号是从输入值中减去时，该反馈称为**负反馈**（negative feedback）。为了控制系统，必须要用负反馈。当反馈信号被加到输入信号上时，将发生**正反馈**（positive feedback）。

2. 控制单元

当收到误差信号时，控制单元将决定采取什么动作。例如，它可能发出操作一个开关或打开一个阀门的信号。当有误差时，控制单元应用的控制策略可能只是提供一个打开或关闭开关的信号。而当房间内有温控器时，信号可能根据误差的大小成比例地打开或关闭阀门。控制方案可能是**硬连接系统**（hard-wired system），其中的控制规划是通过硬件元件连接在一起的方式一直固定的，或者是**可编程系统**（programmable system），控制程序存储在内存单元中，并可通过编程方式改变它。控制器将在第 10 章进行讨论。

3. 校正单元

校正单元在控制过程中发生变化，以对控制条件进行校正或改变。因此，它可能是一个打开加热器的开关，于是增加控制过程的温度，或打开阀门，允许更多的液体进入某一控制过

程。术语**驱动器**（actuator）为校正单元提供功率来执行控制行为。校正单元将在第 7 ~ 9 章进行讨论。

4. 过程单元

这个过程单元就是需被控制的对象。这可能是一个需要温度控制的房间，或需要液位控制的水箱。

5. 测量单元

测量单元产生与某一被控过程的可变条件相关的信号。例如，该信号可能是当达到某一特殊位置的时候开关闭合而产生的开关信号，或者热电偶产生的与温度相关的电动势（e. m. f）。

如图 1-10 所示，由人来控制房间温度的闭环控制系统的各种要素为：

受控变量　—　房间的温度
参考值　—　期望的房间温度
比较单元　—　人对温度的测量值与期望值进行比较
误差信号　—　测量与期望温度之差
控制单元　—　人
校正单元　—　打开炉子的开关
过程　—　通过炉火加热
测量装置　—　温度计

房间温度的自动控制系统可能包括一个恒温元件，它对温度敏感，且当温度低于设定值时打开加热开关，当温度达到设定值时关闭加热开关（见图 1-11）。于是，这一温度敏感开关用于打开加热器。恒温单元兼具比较所期望温度值与实际温度值，以及控制开关操作的功能。在通常情况下，控制系统中的单元能兼具多种功能。

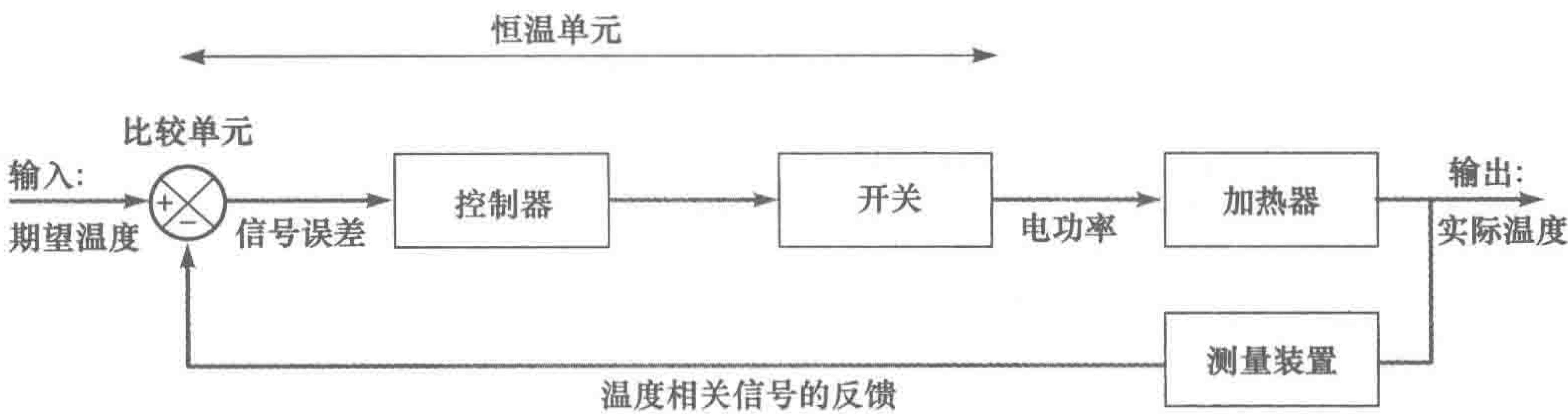


图 1-11　加热某个房间的闭环控制系统

图 1-12 给出了水箱内液位控制系统的简单例子。液位参考值是通过杠杆臂的初始设置位置来设定的，于是当达到期望的液位时，供水被切断。当水从水箱内流出时，浮子随着水位向下移动，从而引起杠杆转动，使水又向箱体内流动。输入水流持续到球体浮子上升到某一高度，使杠杆移动并切断水流供给为止。这是一个有如下要素的闭环控制系统：

13

受控变量　—　水箱中的液位
参考值　—　浮子和杠杆位置的初始设定
比较单元　—　杠杆
误差信号　—　杠杆位置的实际值和初始设置之间的差
控制单元　—　旋转杠杆
校正单元　—　动态摆动打开或关闭供水
过程　—　水箱中的液位
测量装置　—　浮球和杠杆

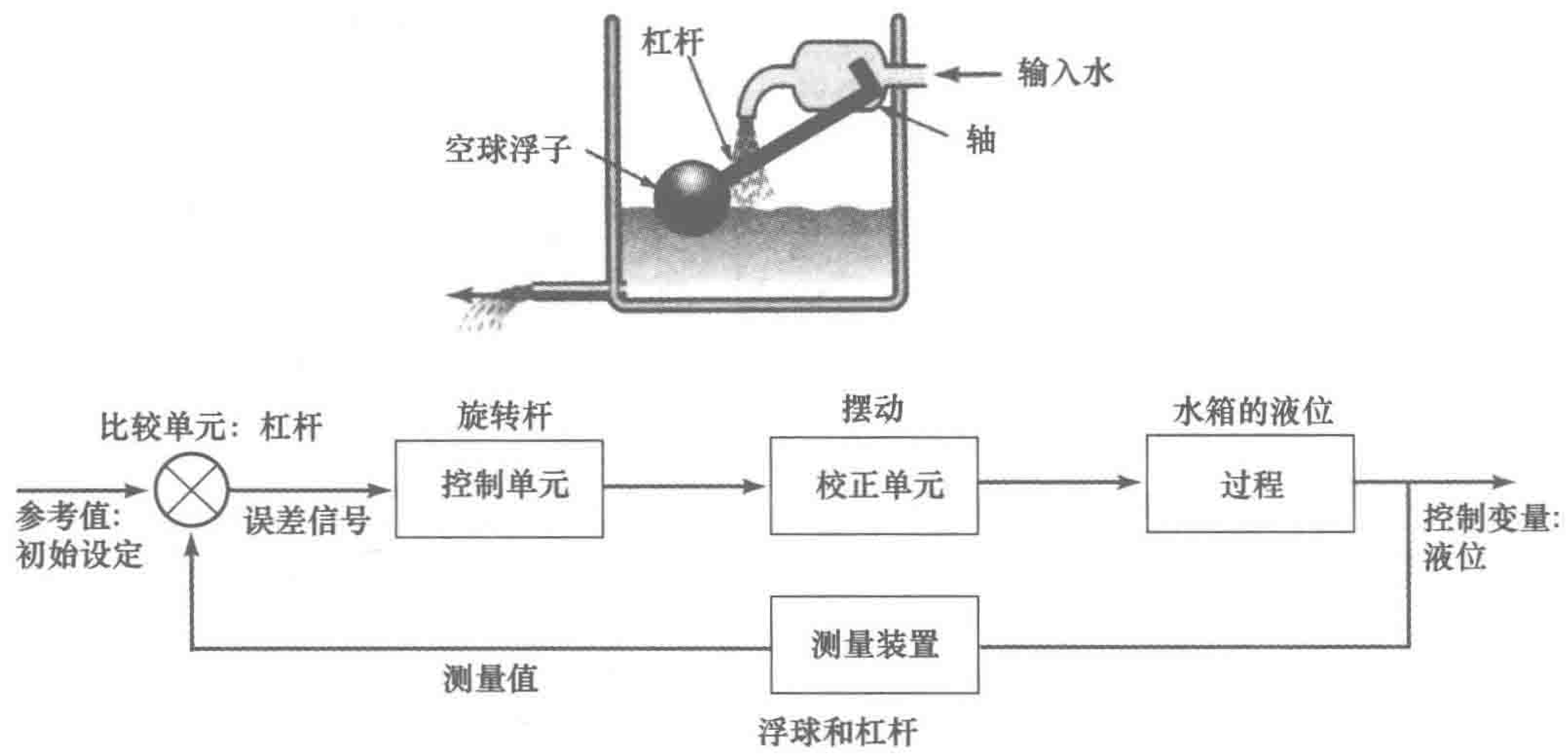


图 1-12　液位的自动控制

以上只是包含机械元件的闭环控制系统的简单例子。然而，我们也可以通过电子控制系统来控制液位。这样，我们可以通过液位传感器提供的电信号，经过合适的信号调理，作为计算机的输入，在计算机内把实际液位值与设定值进行比较，二者之差作为误差信号用于从计算机输出合适的响应。于是，该响应经过合适的信号调理后，用于控制流体控制阀中执行器的运动，进而决定流入水箱的水量。

图 1-13 给出了一个转轴速度自动控制系统的简单例子。电位器用于设定参考值，即将某一电压提供到差分放大器的一个输入端作为期望转速的参考值。差分放大器用于比较参考值与反馈值，并放大二者差分产生的误差信号。然后，将该放大了的误差信号发送给电动机，从而调节转动轴的转速。测速发电机通过一对斜齿轮与电动机轴相连来测量电动机转轴的速度。于是，来自测速发电机的测量信号被反馈到差分放大器的另一端：

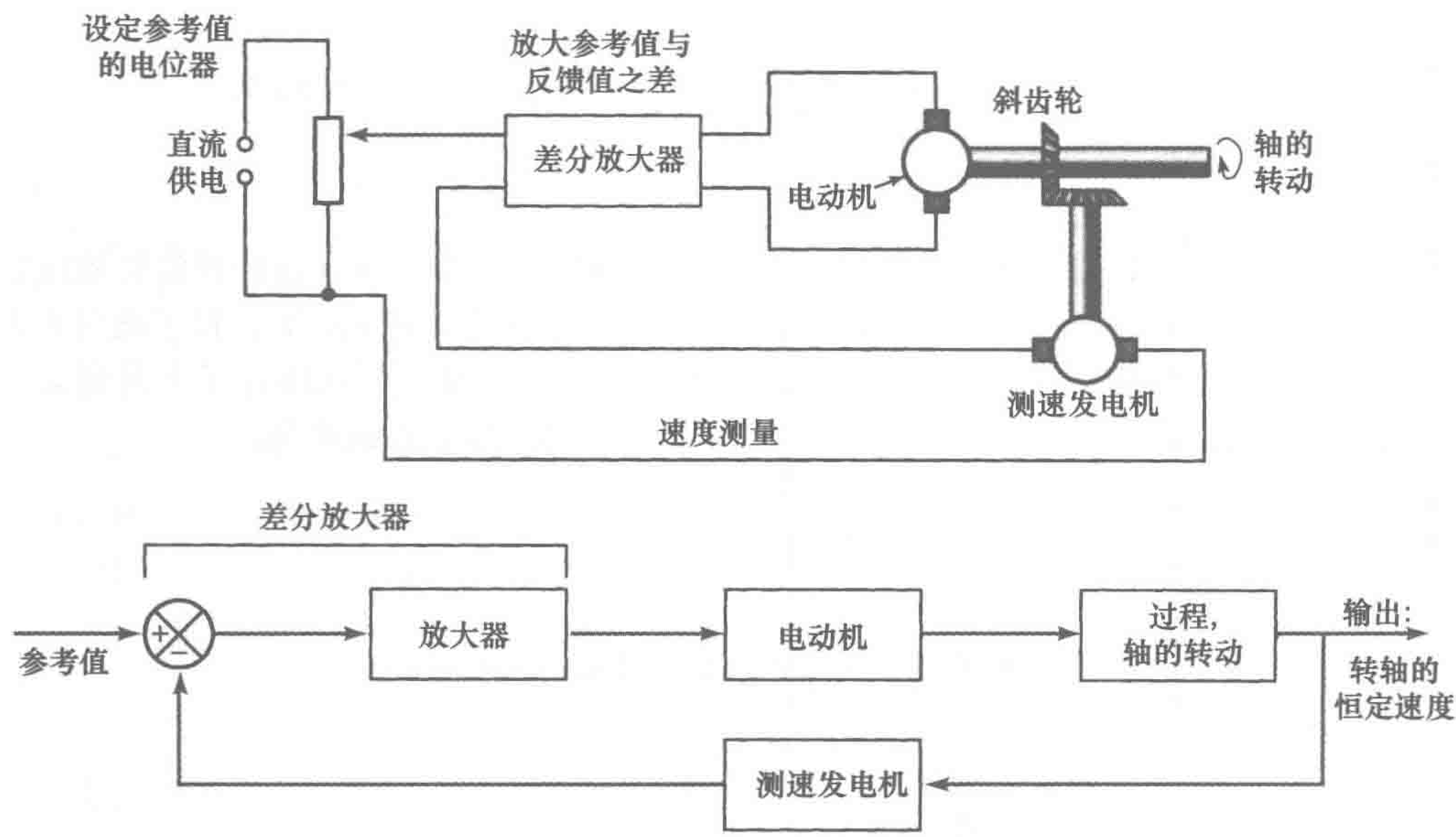


图 1-13　轴速控制

受控变量 — 转轴速度
参考值 — 电位器的滑动端设定
比较单元 — 差分放大器
误差信号 — 电位器的输出与测速发电机的输出之差
控制单元 — 差分放大器
校正单元 — 电动机
过程 — 轴的转动
测量装置 — 测速发电机

14~15

1.5.4 模拟和数字控制系统

模拟系统的所有信号都是时间的连续函数，信号大小表示被测变量的测量尺度（见图 1-14a）。本章到目前所讨论的例子基本上都是这样的系统。而**数字信号**可考虑为一系列的 on/off 信号，数字变量值由 on/off 脉冲序列来代表（见图 1-14b）。

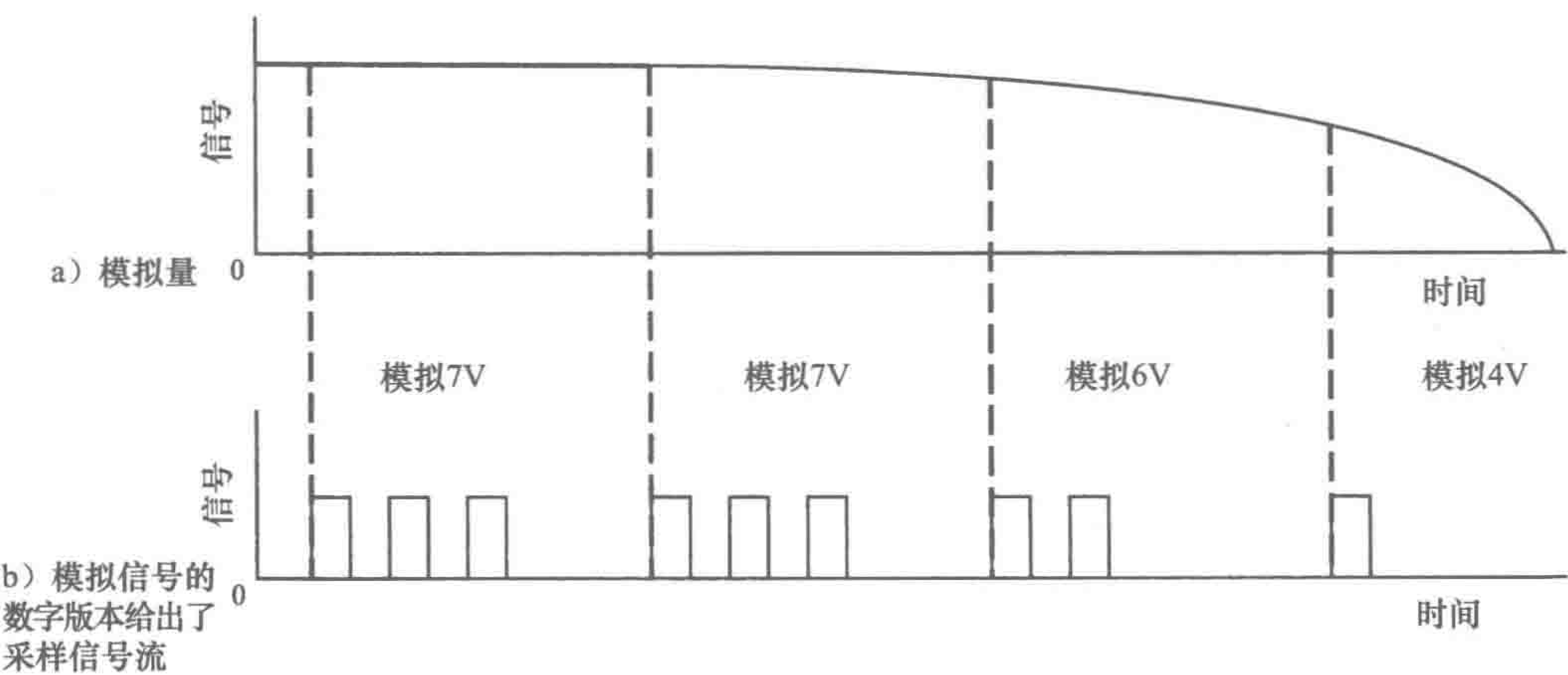


图 1-14 信号

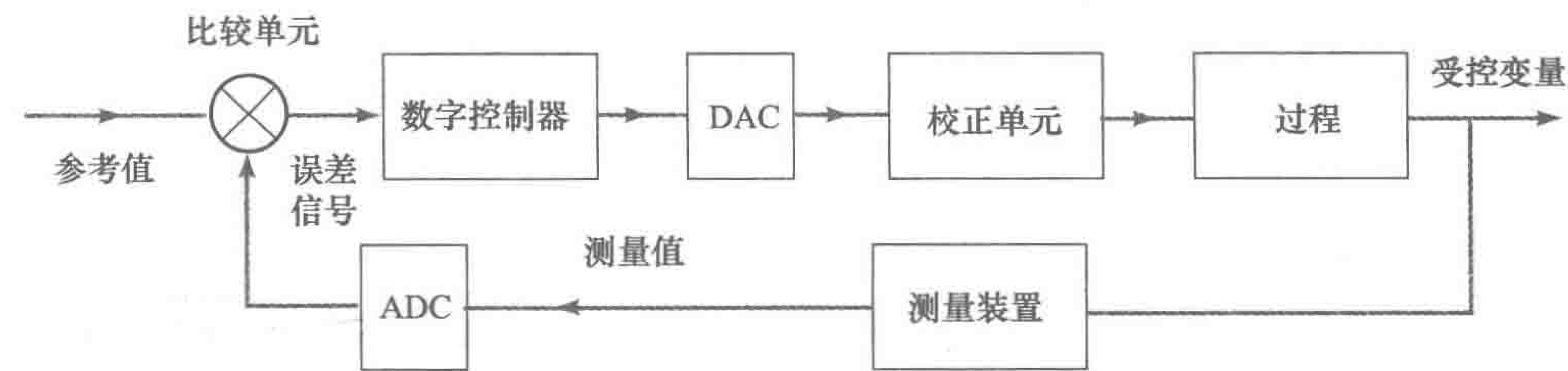
数字信号可用于代表连续的模拟信号。在某个特定的时刻模拟信号被采样，然后每一采样值被有效地转换成数字值，即某一特殊的数字信号序列。例如，对于三位数字信号，可以有如下数字序列：

无脉冲，无脉冲，无脉冲，代表模拟信号 0V；
无脉冲，无脉冲，一个脉冲，代表模拟信号 1V；
无脉冲，一个脉冲，无脉冲，代表模拟信号 2V；
无脉冲，一个脉冲，一个脉冲，代表模拟信号 3V；
一个脉冲，无脉冲，无脉冲，代表模拟信号 4V；
一个脉冲，无脉冲，一个脉冲，代表模拟信号 5V；
一个脉冲，一个脉冲，无脉冲，代表模拟信号 6V；
一个脉冲，一个脉冲，一个脉冲，代表模拟信号 7V。

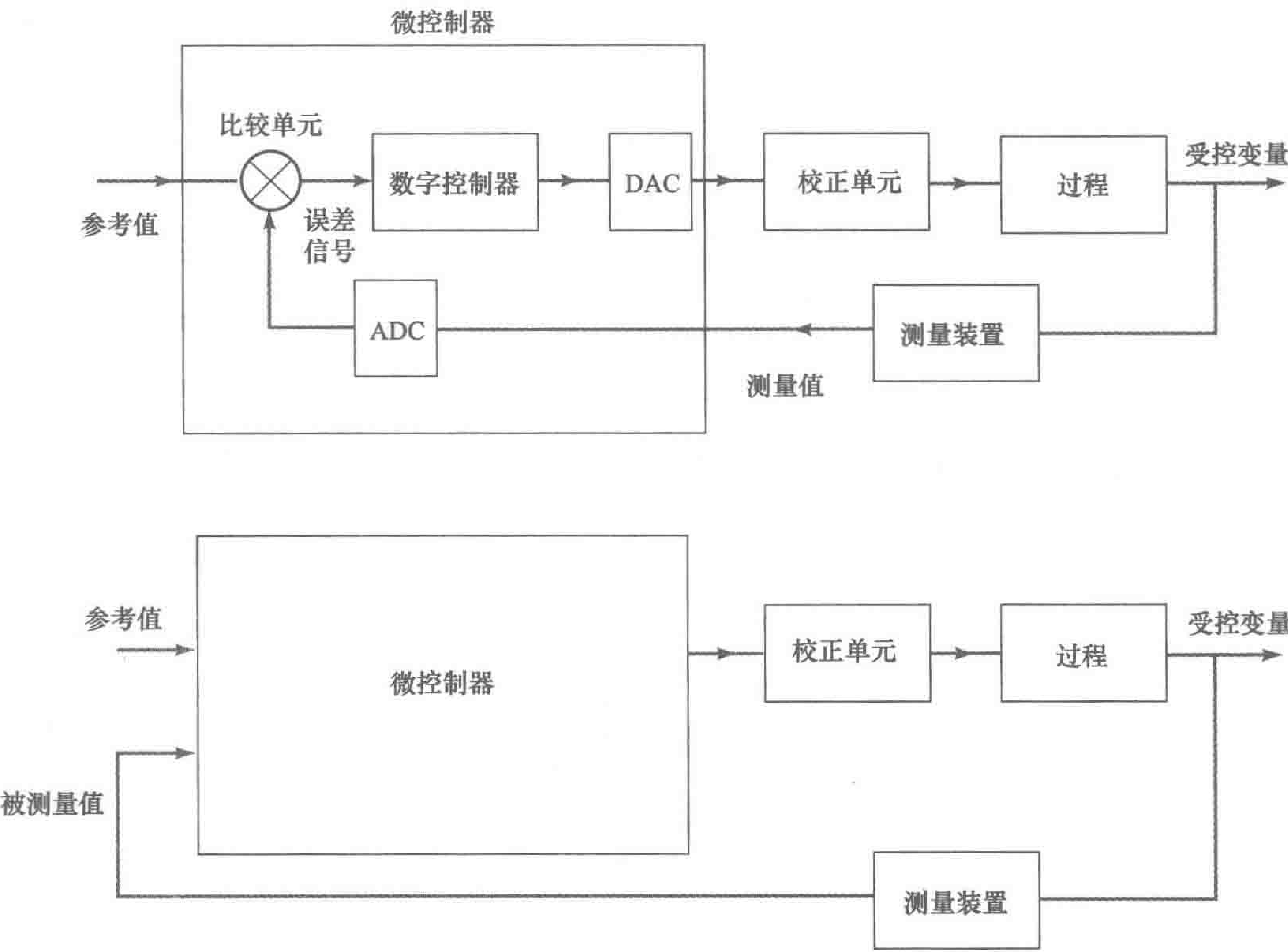
自然界中大多数情况下被控制的量是模拟量，而且它们是控制系统的输入和输出。例如，温度输入和加热器的输出，数字控制系统的必要特征是真实世界的模拟输入必须被转换成数字形式，而且数字输出必须被转换成真实世界的模拟形式。这包括对于输入的模数转换器（ADC）和对输出的数模转换器（DAC）的应用。

与图 1-10 中给出的模拟闭环系统相比，图 1-15a 给出了一个数字闭环控制系统的基本组成单元。参考值或设定值可以是来自键盘的输入。为了能向数字控制器提供来自模拟检测系统的

数字信号或数字信号输出能被转换成模拟信号来操作校正单元，模数转换（ADC）和数模转换（DAC）单元被包含在闭环中。加入 ADC 和 DAC 好像增加了控制系统的复杂度，但却带来了非常重要的优点：数字操作能通过程序（即一系列存储的指令）来控制；信息存储容易，精度更高；数字电路被噪声影响的可能性更小；更容易设计。



a）一个数字闭环控制系统的基本要素



b）一个微控制器控制系统

图 1-15　数字闭环系统

16～17

数字控制器可能是运行一段软件程序的数字计算机，从而实现期望的行为。术语控制算法（control algorithm）用来描述需要解决控制问题的一系列步骤。用于数字控制的控制算法可用如下步骤来描述：

```
读参考值,即期望值。
读实际设备的 ADC 输出。
计算误差信号。
计算期望的控制器输出。
```

将控制器输出发送到 DAC。
等待下一个采样周期。

然而，很多应用并不需要贵重的计算机，微型芯片就可以满足需要了。于是，在机械电子学的应用中，微型控制器经常用于数字控制。微型控制器是一种集成了诸如存储器、ADC、DAC 等单元的微型处理器；这些单元能直接与受控设备相连，图 1-15b 给出了具体的连接安排。于是，控制算法可能为：

读参考值，即期望值。
读实际设备的输出到 ADC 输入端口。
计算误差信号。
计算期望的控制器输出。
将控制器输出发送到 DAC 输出端口。
等待下一个采样周期。

例如，一个房间温度的自动控制系统就可以是一个数字控制系统。它包含输出模拟信号的温度传感器，经过合适的信号调理后变成数字信号，再被输入到数字控制器。在数字控制器内其与设定值比较，从而产生误差信号。然后，误差信号被数字控制器处理后输出数字信号，经过适当的信号调理，给出等效的模拟信号来控制加热器，因而室温得到了控制。这样的系统能很容易地通过编程实现在一天不同的时间给出不同的温度。

作为数字控制系统的进一步说明，图 1-16 给出了一种电动机可以采用的速度数字控制系统的形式。可以把该图与图 1-13 中的模拟系统进行比较。

数字控制器的软件需要能实现如下功能：

从输入端口读数据。
进行内部的数据传输和数学运算。
发送数据到输出端口。

此外，它还易于决定何时将执行控制程序。

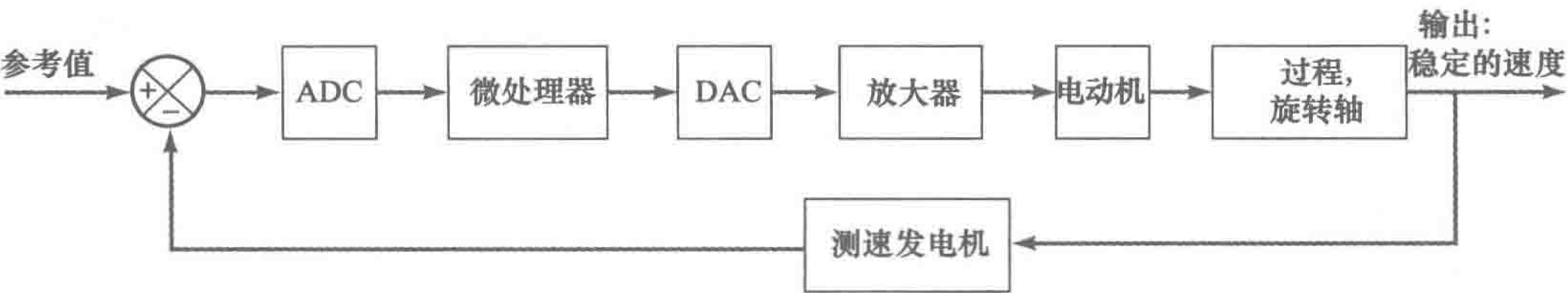


图 1-16 转轴速度控制

18

于是，我们可以让程序等待 ADC 采样时间到来，当有采样输入时触发一定的动作。针对这些采样事件，在这种情况下可采用**轮询法**（polling）使程序重复地检查输入端口。于是我们有：

检查输入信号的输入端口。
无信号，则什么也不做。
检查输入信号的输入端口。
无信号，则什么也不做。
检查输入信号的输入端口。
有信号，则从输入端口读数据。
执行内部数据传输和数学运算。
发送数据到它的输出端口。
检查输入信号的输入端口。
无信号，则什么也不做。
等等。

另一种替换轮询的方法是**中断控制**。程序不是在一直检查输入端口，当输入时间到时，接收一个输入信号。该信号可能来自于一个外部时钟，该时钟在每次 ADC 采样时给出一个信号。于是有：

```
没有外部时钟信号。
什么也不做。
当输入发生时,信号从外部时钟输入。
从它的输入端口读数据。
执行内部数据传输和数学运算。
发送数据到它的输出端口。
等待来自外部时钟的另一个信号。
```

1.5.5　顺序控制器

为了控制生产过程和操作序列，在很多情况下控制任务是按照顺序要求，在特定的预设时间或变量值下，通过开关来实现的。例如，步骤 1 完成后，步骤 2 开始。步骤 2 完成后，步骤 3 开始，等等。

当控制行为是严格按照时间或事件驱动的序列来安排时，我们就应用**顺序控制**（sequential control）方法。这一控制可通过一系列的继电器或凸轮开关组成的电路来实现。这些电路要连线成合适的方式，才能给出期望的顺序。目前，这一硬件电路可能逐步被微处理器控制系统代替。于是，通过软件编程，就可实现顺序控制。

用家用洗衣机来说明顺序控制。洗衣机的一些操作必须按照正确的顺序执行。当用冷水浸泡桶内的衣服时，处于预洗涤周期。接着，当给热水时，处于主洗涤周期。然后，衣服在桶内用冷水漂洗几次，处于漂洗周期。最后，通过旋转使衣服脱水。这些操作的每一项都包含一系列的步骤。例如，预洗涤周期包括打开阀门使洗衣桶内注水到期望的液位，关闭阀门，打开洗衣桶的电动机使洗衣桶旋转一段时间，打开泵把桶内的水抽空。这一系列的操作称为**程序**（program），程序中的一系列指令是事先定义好的，并写入所用的控制器。

19

图 1-17 给出了基本洗衣机系统的组成元件的粗略想法。过去经常应用的洗衣机控制器是机械的，它包含一系列的凸轮操作开关，即机械开关系统，可以很容易地给出各种操作序列进行调节。

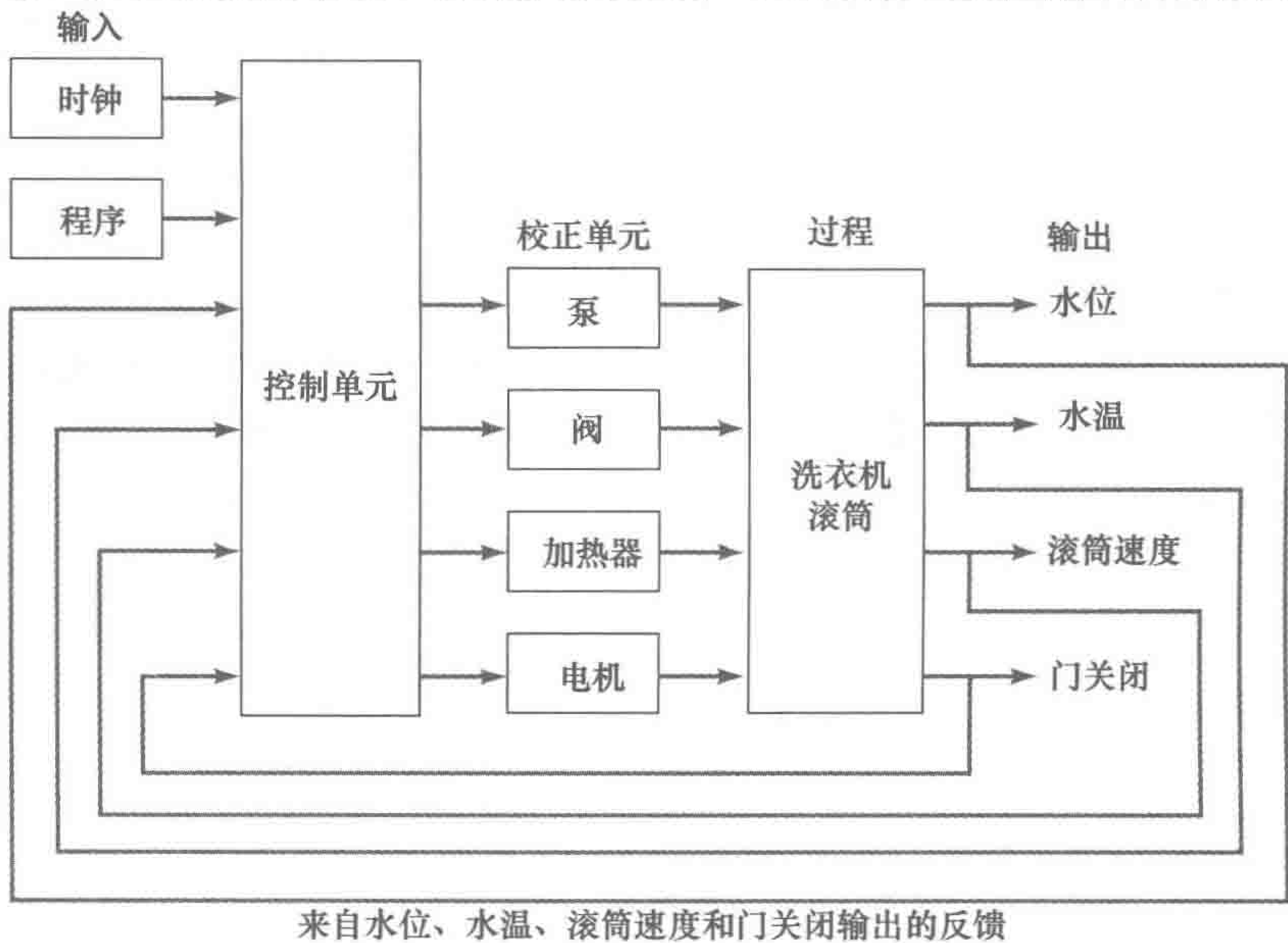


图 1-17　洗衣机系统

图 1-18 给出这样一种开关的基本原理。当机器通电时，小电动机的转轴慢慢旋转，电动机轴的转动量与时间成比例。电动机的转动带动控制器的凸轮转动，依次操作电气开关，并以正确的顺序接通电路。凸轮的轮廓决定了何时操作开关。这样，凸轮的轮廓决定了操作的程序，并存储在了机器中。在特定的洗衣程序中应用的指令与指令序列是由所选择的一系列凸轮决定的。对于现代的洗衣机来说，控制器是一种微处理器。操作序列不是由凸轮的机械排列提供的，而是由软件程序实现的。这种由微处理器控制的洗衣机被认为是机械电子学方法的一个典型例子，其中机械系统与电子控制已集成在了一起。其结果是体积庞大的机械系统被更加紧凑的微处理器所代替。

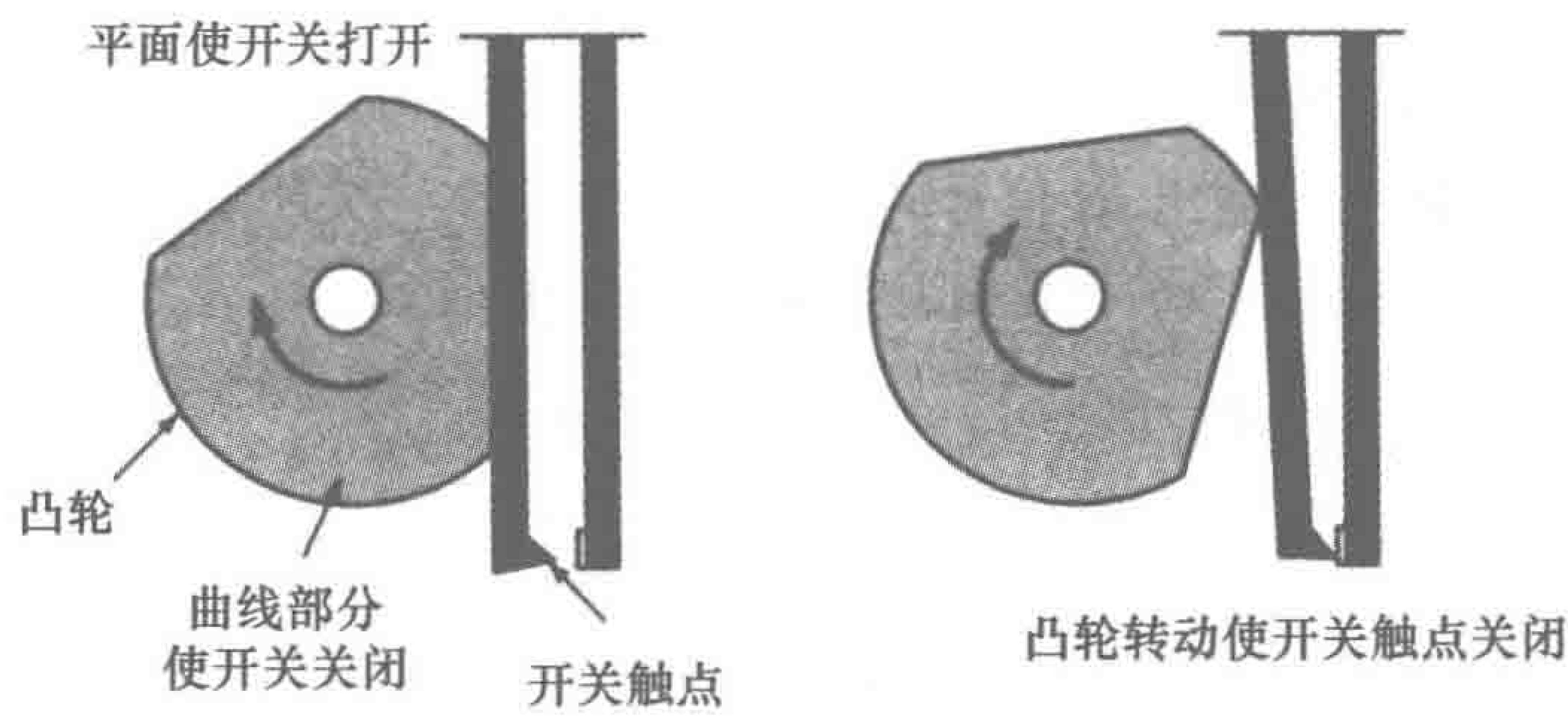


图 1-18 凸轮操作的开关

对于预洗涤周期，当有电流通过时打开电磁阀，当电流停止时电磁阀关闭。操作开关的凸轮的轮廓或微处理器的输出决定了电磁阀向桶内放入冷水的时间长短。然而，由于在洗衣桶内需要特定的液位，因此当达到要求的位置时，在允许的时间内，需要另外的机械来停止供水。当液位达到了预设位置时，需要一个传感器给出一个输入信号，然后微处理器发出的输出信号切断电磁阀的电流。在凸轮控制阀门的情况，传感器驱动一个开关，从而关闭向洗衣机滚筒内放水的阀门。当这一事件完成后，微处理器或凸轮的转动就启动泵，抽空滚筒内的水。

当预洗涤程序完成后，微处理器给出一个输出，开始主洗涤周期。在凸轮操作的情况下，当预洗涤周期完成后，凸轮的轮廓使主洗涤开始。它接通进入电路的电流，打开阀门，使冷水进入滚筒。时刻检测液位。当达到期望的位置时，停止进水。然后，微处理器或凸轮提供电流激活一个开关，向电加热器施加更大的电流来加热水。当温度传感器检测到水温达到预设值后，切断加热电流。于是，微处理器或凸轮打开滚筒电动机使滚筒旋转。旋转的时间是由微处理器或凸轮的轮廓决定的。然后，微处理器或凸轮使水泵通电，使滚筒内的水排空。

当信号打开阀门时，冷水进入洗衣机滚筒，于是漂洗操作开始。然后，阀门关闭，操作电动机转动滚筒，再操作泵从滚筒中抽空水，然后重复上述过程几次。

当微处理器或凸轮打开滚筒电动机时，处于最后的操作阶段。此时电动机比漂洗有更高的转速，从而使滚筒内的衣服快速旋转。

1.6 可编程逻辑控制器

在很多简单系统中，有一个嵌入式微控制器，即存储器和微处理器都集成到一个芯片上，并且已针对所关心的任务进行了特定的编程。一种更有适应能力的形式是**可编程逻辑控制器**（Programmable Logic Controller，PLC）。这是一种基于微处理器的控制器，它应用可编程存储器来存储指令，执行逻辑、顺序、定时、计数和算数运算等功能，实现对事件的控制，并且能很容易针对不同的任务重新编程。图 1-19 给出了可编程逻辑控制器的控制行为，输入信号来自闭合的开关，而由程序来决定控制器应该怎样对输入做出响应，并给出什么样的输出。

20 ~ 21

可编程逻辑控制器在需要开关控制的工业场合得到了广泛应用。例如，它们可能用于这样一个过程控制中：给一个箱体先充入液体，然后在排空前将其加热到指定的温度。控制顺序如下：

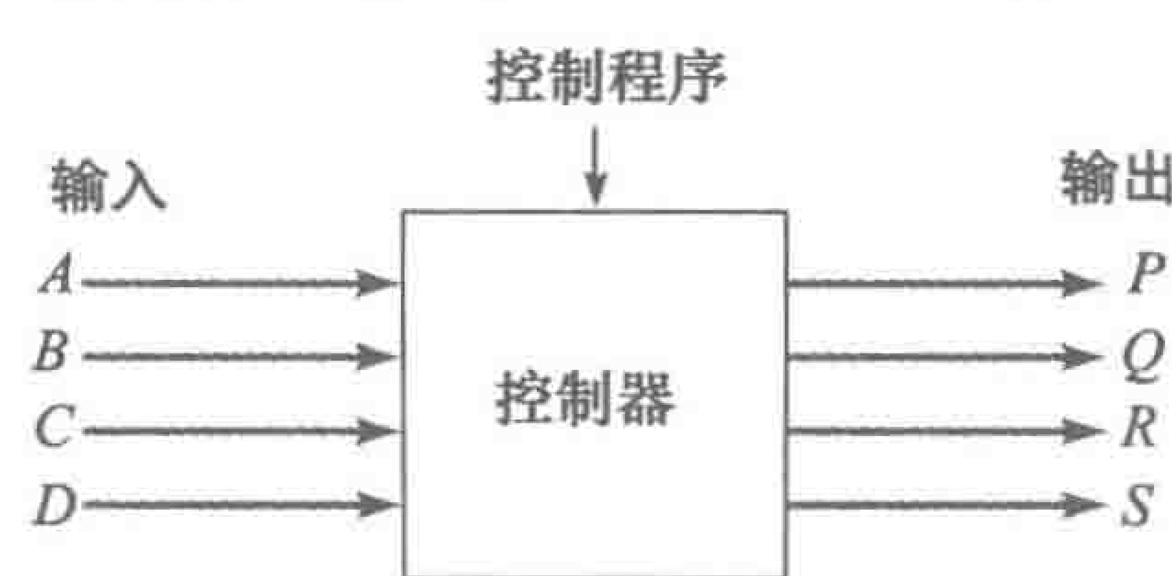


图 1-19 可编程逻辑控制器

1）打开一个泵，使液体进入箱体。

2）当液位传感器输出 on 信号时，表明液位已经达到了期望的位置，关闭泵。

3）打开加热器。

4）当温度传感器输出 on 信号时，表明温度已经达到了期望值，关闭加热器。

5）打开泵，从容器中抽空液体。

6）当另一个液位传感器输出 on 信号时，表明箱体已空，关闭泵。

关于可编程逻辑控制器的详细讨论和应用例子，请见第 14 章。

1.7 机电一体化系统的例子

机械电子学把传感器与测量系统、嵌入式微处理器系统、驱动器与工程设计技术融为一体。下面的机电一体化系统的例子说明了基于微处理器的系统不但能执行原来通过“机械”的方式实现的任务，还能实现以前难以自动实现的任务。

1.7.1 数码相机与自动对焦

数码相机一般都有一个自动对焦控制系统。便宜的相机应用的基本系统是开环系统（见图 1-20a）。当摄影师按下快门时，相机前面的传感器向拍照对象发出红外光脉冲。这一红外脉冲从拍照对象反射到相机，并被相机上相同的变送器获得。通常，景物与相机距离 1m，红外光的往返时间约 6ms。输出和返回脉冲的时间差被检测出来，并被输入到微处理器。微处理器里有一系列的值存储在它的存储器里，通过计算给出合适的输出，使镜头转动到焦点对准的位置。当距离增大时，返回脉冲就非常弱，因此这类自动对焦相机只能用于距离小于 10m 的情况。于是对于更大的距离，微处理器给出的输出使镜头转到极限的设定位置。

22

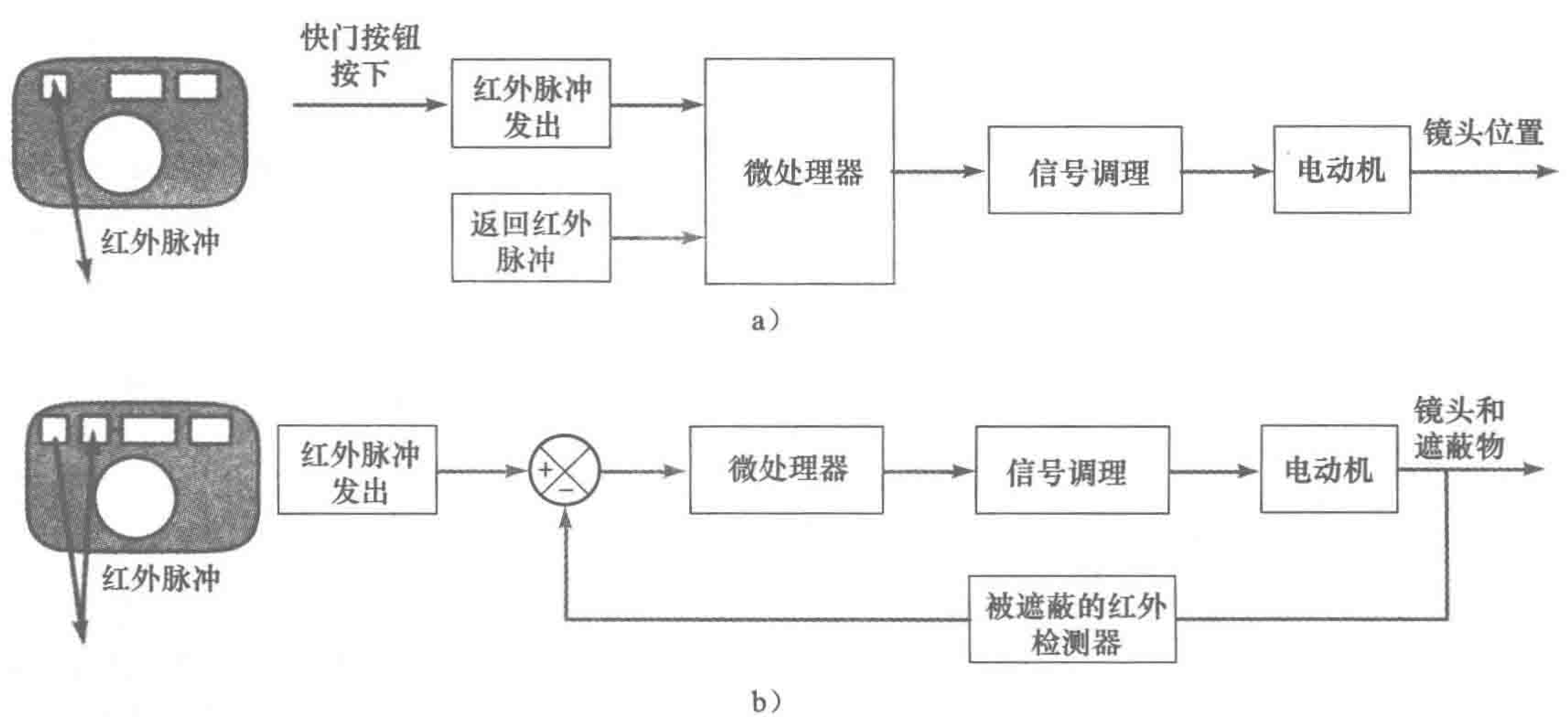


图 1-20 自动对焦

更昂贵的相机使用的系统包括三角测量系统（见图 1-20b）。发出红外脉冲和检测反射脉

冲不是通过一个相同的负责发送的变送器来实现的，而是通过另外一个变送器。然而，初始状态下这个变送器上有一个遮蔽物。当微处理器给出输出使镜头移动时，这个变送器上的遮蔽物同时跟着移动。这个遮蔽物有一个狭缝，可在变送器的表面移动。镜头和狭缝一直移动，直到返回的脉冲能通过狭缝并打在变送器上为止。于是，发自变换器的输出使微处理器停止镜头的运动，并最终给出焦点的位置。

1.7.2 发动机管理系统

汽车的发动机管理系统负责管理发动机的点火与燃料供给。四冲程内燃发动机有一些缸体，每个缸体有一个活塞连接到一个公共曲轴，并执行四冲程的操作序列（见图 1-21）。

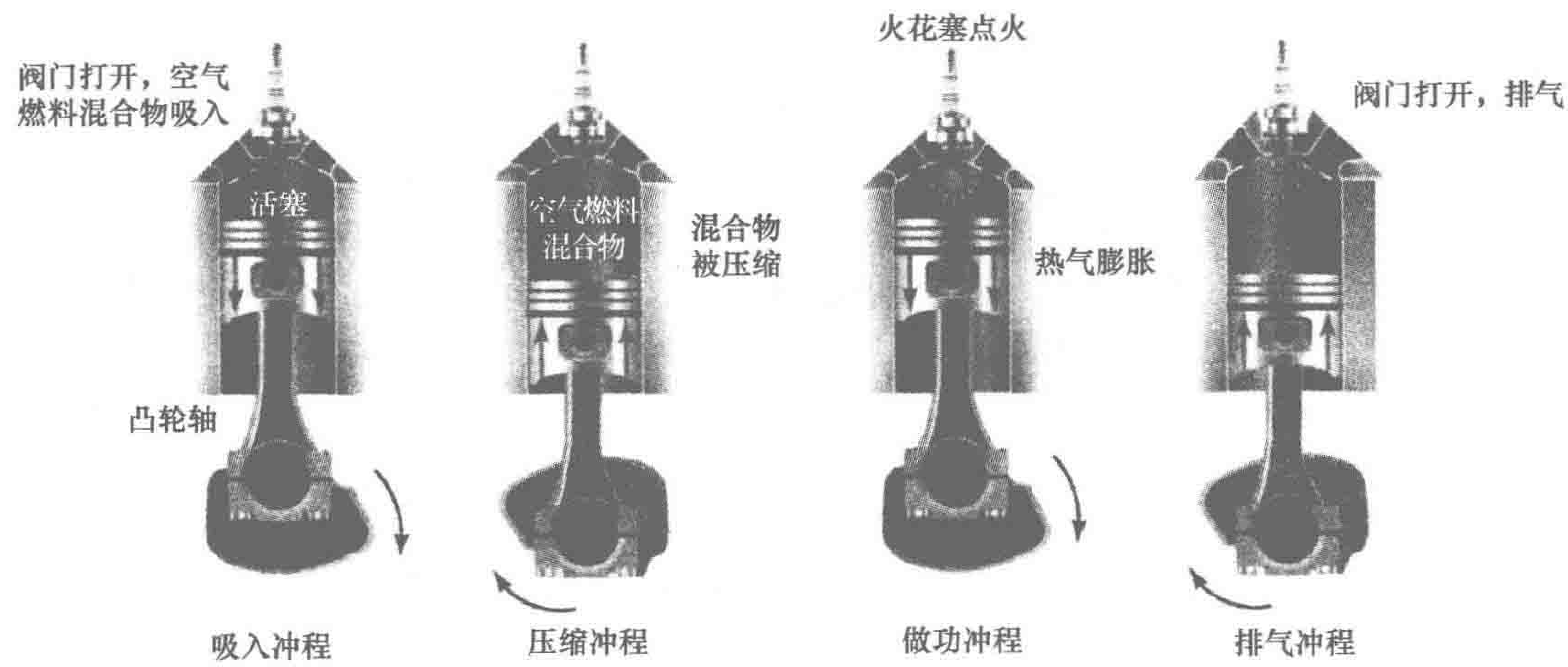


图 1-21　四冲程的操作顺序

当活塞向下运动时，阀门打开，空气燃料混合物吸入缸体。当活塞再向上移动时，阀门关闭，空气燃料混合物被压缩。当活塞接近缸体顶部时，火花塞点燃混合物，使热气膨胀。膨胀引起活塞重新向下运动并做功，上述周期重复进行。由于每一缸体的活塞都连接到了一根曲轴上，且做功冲程发生在不同的时间段，所以转动曲轴的做功是连续的。

发动机的做功和速度是由改变点火时间和空气燃料混合物来控制的。对于现代的汽车发动机来说，这是通过微处理器完成的。图 1-22 给出了微处理器控制系统的基本要素。对于点火

23

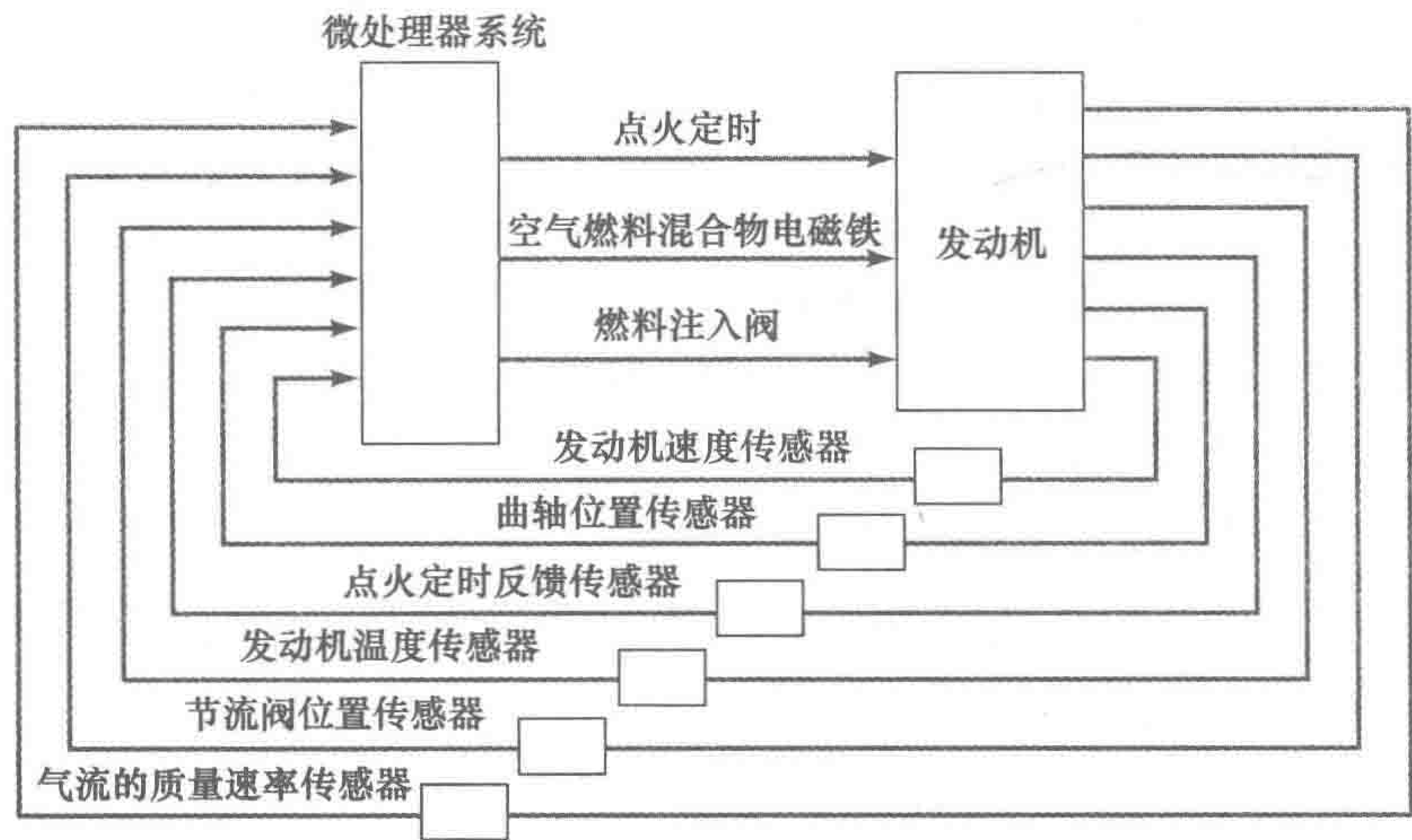


图 1-22　发动机管理系统的基本要素

时间，曲轴驱动时间轮和分配器使每个火花塞依次产生电接触。时间轮产生的脉冲指示曲轴的位置。于是，微处理器调整时间，发出高压脉冲给分配器，使脉冲产生在合适的时刻。在吸气冲程，为了控制空气燃料混合物进入缸体的量，根据发动机的温度和节流的位置，微处理器改变电磁铁打开进气阀门的时间。被注入气流中的燃料量由气流的质量速率传感器或来自其他的测量计算输入决定，于是微处理器给出输出来控制燃料注入阀。注意，上述只是发动机管理的非常简单的说明。更详细的说明见第 24 章。

24

1.7.3　MEMS 和汽车安全气囊

MEMS（Micro Electro Mechanical System，**微机电系统**）是一种构建在半导体芯片上的机械装置，尺寸从 20μm ~ 1mm，且由 0.001 ~ 0.1mm 大小的部件组成。MEMS 通常由微处理器和微传感器、微驱动器等部件组成。MEMS 可在微米尺度感知、控制和激励机械过程。这种 MEMS 芯片已经得到了越来越广泛的应用，下面给出了说明。

在汽车发生猛烈碰撞时，设计的安全气囊膨胀，从而缓冲对车辆乘员的冲击影响。安全气囊传感器是一种集成了微机械元件的 MEMS 加速度计，对快速的减速做出响应。图 2-9 给出了广泛使用的 ADXL-50 装置的基本细节。快速的减速引起 MEMS 加速度计中的电容发生改变，被 MEMS 芯片中的电子装置检测到，并激励气囊控制单元对气囊点火。于是，气囊控制单元触发气体发生器推进剂点燃，使尼龙织物袋迅速膨胀（见图 1-23）。当车辆乘员的身体发生碰撞并挤压充气气囊时，气体将以可控的方式从小孔排出，于是缓冲了冲击。从一开始撞击到整个安全气囊的展开和膨胀过程大约是 60 ~ 80ms。

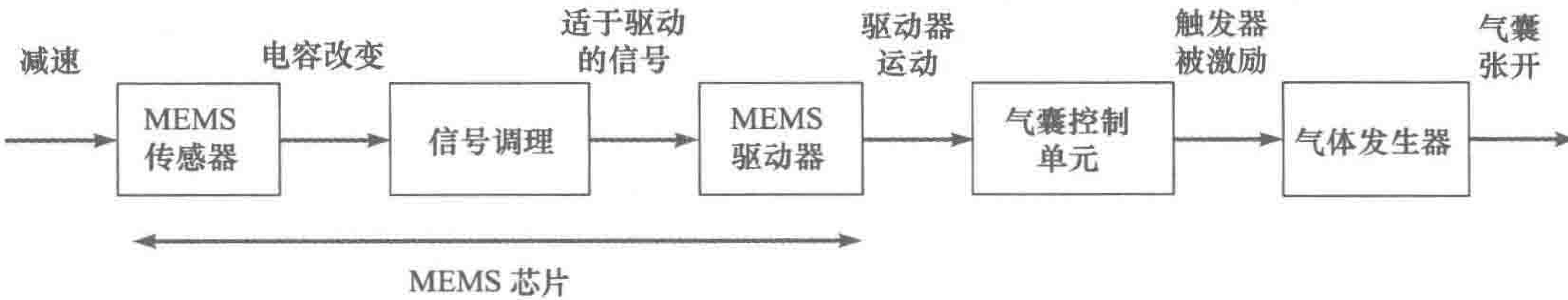


图 1-23　气囊控制系统

本章小结

机械电子学是一种协调同步发展起来的技术，是在产品的设计与制造过程中的机械工程、电子学和智能计算机控制的集成。它把几种技术涵盖在一起：机械工程、电子工程、电气工程、计算机技术和控制工程。机械电子学向工程师提供了一种以新的眼光看问题的机会，不只是以机械原理的角度看问题，而是以多种技术的眼光来看问题。电子仪器等不应该被看作是已有的机械硬件上类似螺栓的零件。机械电子方法需要从设计阶段就开始采用。

机电一体化系统通常包括**嵌入式微处理器**。嵌入式系统用于控制一定的功能，但不能被终端用户通过计算机等相同的方式修改程序。这样，对于嵌入式系统来说，用户不能通过增加或替换软件来改变系统的功能。

25

系统可被认为是有输入/输出的盒子或框图。我们关心的不是盒子内发什么，而只关心输入和输出之间的关系。

为了预测当有输入时系统如何表现，需要设计与输出/输入相关的模型，于是对于给定的输入，我们就能确定输出将如何随着时间变化。

通常，**测量系统**有三个基本的组成单元：传感器、信号调理器与显示器。

控制系统的基本形式有两种：**开环和闭环**。闭环存在反馈，于是系统包括比较单元、控制

单元、校正单元、过程单元和包含测量单元的反馈单元。

习题

1.1　请识别水银玻璃温度计和弹簧压力表两种测量系统中的传感器、信号调理器和显示单元。

1.2　请解释开环与闭环控制的区别。

1.3　请识别在恒温控制电子加热器的控制系统中出现的各种要素。

1.4　一个浴缸的水温自动控制系统包含一个输入到差分放大器的参考电压。放大器与一个继电器相连，打开或关断水中加热器的供电。负反馈是通过测量系统向差分放大器输入电压来实现的。请画出该系统的框图，并说明误差信号是如何产生的。

1.5　请说明可编程逻辑控制器的功能。

1.6　请说明顺序控制是什么意思，通过一个例子说明你的回答。

1.7　请说明洗碗机顺序控制的步骤。

1.8　请对比传统的手表设计与包含微处理器的机械电子产品的设计区别。

1.9　请对比“双金属恒温器”与“微处理器”的室内中央加热系统的控制系统的区别。

26

第二部分　传感器与信号调理

第2章　传感器和变送器

本章目的

通过本章的学习，读者将能够：

- 用测量范围、量程、误差、精度、灵敏度、迟滞和非线性误差、重复性、稳定性、死区、分辨率、输出阻抗、响应时间、时间常量、上升时间和建立时间等术语描述常用的传感器性能。
- 评估用于测量位移、位置和微位移、速度和运动、力、流体压力、流体流动、液位、温度和光强的传感器。
- 解释当机械开关用于输入数据时产生的跳动问题以及如何解决。

2.1　传感器和变送器简介

术语**传感器**（sensor）是指能够产生与被测量的量相关信号的元件。所以就电阻温度元件来说，被测量的量是温度，然后传感器把温度输入转化为电阻变化。**变送器**（transducer）这个术语通常用于代替传感器（sensor）这个术语。变送器定义为当受到一些物理变化时会产生相关变化的元件。因此传感器属于变送器。然而，一个测量系统在把信号从一种形式转化为另一种形式的时候除了传感器之外还可以用变送器。如果一个传感器/变送器的输出量是连续变化的模拟量，同时输出量的大小和被测量的变化大小是成比例的，则称为**模拟式**（analogue）传感器/变送器。如果它们给出的输出本来就是数字形式，即一系列的开关信号，用于说明数字的值与被测量变量之间的关系，则称为**数字式**（digital）传感器/变送器。

本章是关于变送器的，尤其是那些用做传感器的变送器。给出那些用于描述变送器性能术语的定义，同时也会讨论在工程中常用的变送器的例子。

2.1.1　智能传感器

27~29

一些传感器和它们的信号调理部分都整合到了一个封装内。这种集成的传感器仍然需要进一步的数据处理。但是，把传感器、信号调理和微处理器集成到一起还是有可能的。我们把有这种设计的传感器称为**智能传感器**（smart sensor）。智能传感器能够补偿随机误差，适应环境的变化，自动计算测量的精度，把非线性调整为线性输出，自行校正并自行进行错误诊断。

这种传感器有它们自己的标准，IEEE 1451，因此符合这种标准的传感器可以“即插即用”，用标准方式存储和传输数据。信息是用变送器电子数据手册（TEDS）的形式存储的，一般存储在电可擦只读存储器（EEPROM）中，并且每一个器件带有标志，并能给出校准的数据。

2.2　性能术语

以下术语用于定义变送器和整个测量系统的性能指标。

1. 测量范围和量程

变送器的测量范围（range）定义了输入量能够变化的范围。量程（span）是输入的最大量减去最小量。例如，一个测力传感器可能的测量范围是0～50kN，而它的量程是50kN。

2. 误差

误差（error）是测量值与真实值的差：

$$误差 = 测量值 - 真实值$$

因此，如果一个测量系统给出温度的读数是25℃，它的实际温度是24℃，则误差是+1℃。如果实际温度是26℃，那么误差是-1℃。一个传感器可能给出电阻的变化是10.2Ω，而真实的变化应该是10.5Ω，那么误差为-0.3Ω。

3. 精度

精度（accuracy）表示的是一个测量系统测出的值可能出错的程度大小。它因此是所有可能出现的误差和已经校正过的变送器的精度的总和。比如说，一个温度测量仪器，假如它的精度是±2℃。这意味着仪器给出的读数在真值加上2℃和减去2℃之间。精度通常表示为满量程输出或者满偏值的百分比。当测量系统的显示输出几乎全在一个圆形或线性尺度上时，产生了满偏百分比这个术语。例如，说一个传感器的精度是满量程输出的±5%。则如果这个传感器的测量范围是0～200℃，那么其读数在真值加上和减去10℃之间。

4. 灵敏度

灵敏度（sensitivity）表示单位输入量变化时输出量的变化大小，就是输出/输入。例如，30一个电阻温度计的灵敏度为0.5Ω/℃。灵敏度也通常用于表示对输入的灵敏度而非对被测量的灵敏度，比如说环境变化。因此有传感器对环境温度变化的灵敏度或者对电源电压波动的灵敏度。例如，可以说一个测量压力的传感器有对环境每摄氏度温度变化±0.1%的灵敏度。

5. 迟滞误差

在变送器的值连续增大或者连续减小时，对应同一个测量值的输出信号却不同，这个效应称为迟滞。图2-1表示的是在增大和减小过程中输出量的最大差值，称为迟滞误差（hysteresis error）。

6. 非线性误差

对于许多变送器都假定在工作范围内输入和输出之间是线性的，即输入/输出的关系是一条直线。然而，几乎没有传感器是真正线性的，因此线性假设将会引入误差。非线性误差（non-linearity error）定义为实际输出与假设的线性输出之间的最大差值。量化非线性误差有很多不同的方法。这些方法的区别在于如何找到那条用于计算误差的直线。一种方法是画出连接测量范围头尾两个点的直线；另一种方法是用最小二乘法找到最优拟合曲线，使所有数据点都在合理的误差范围内；另一种方法是用最小二乘方法找到通过零点的最优拟合曲线。图2-2表示的是这三种方法及它们对所述非线性误差的影响。非线性误差一般表示为满量程输出的百分比。例如，一个测量压力的变送器可以说它有满量程±0.5%的非线性误差。

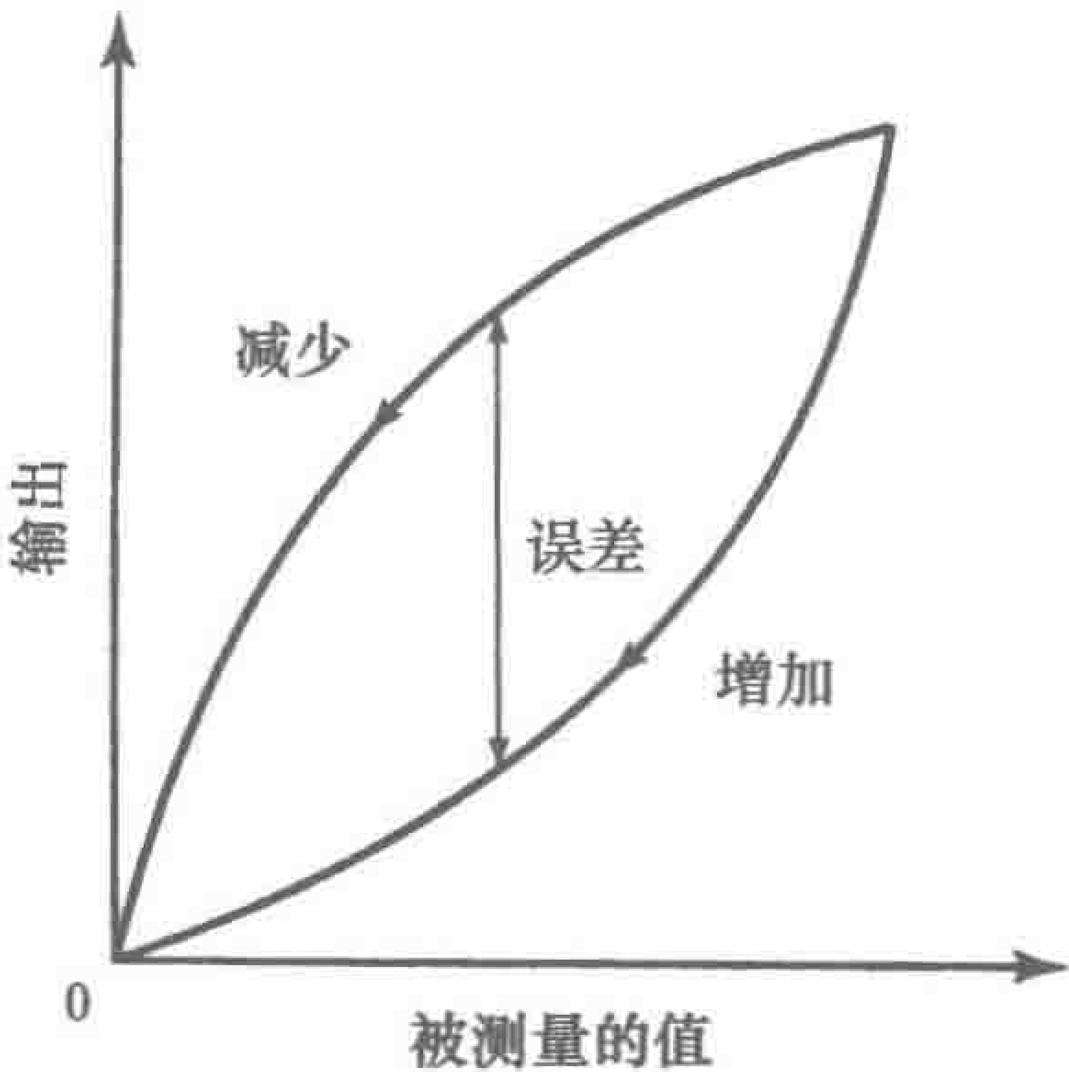


图2-1　迟滞

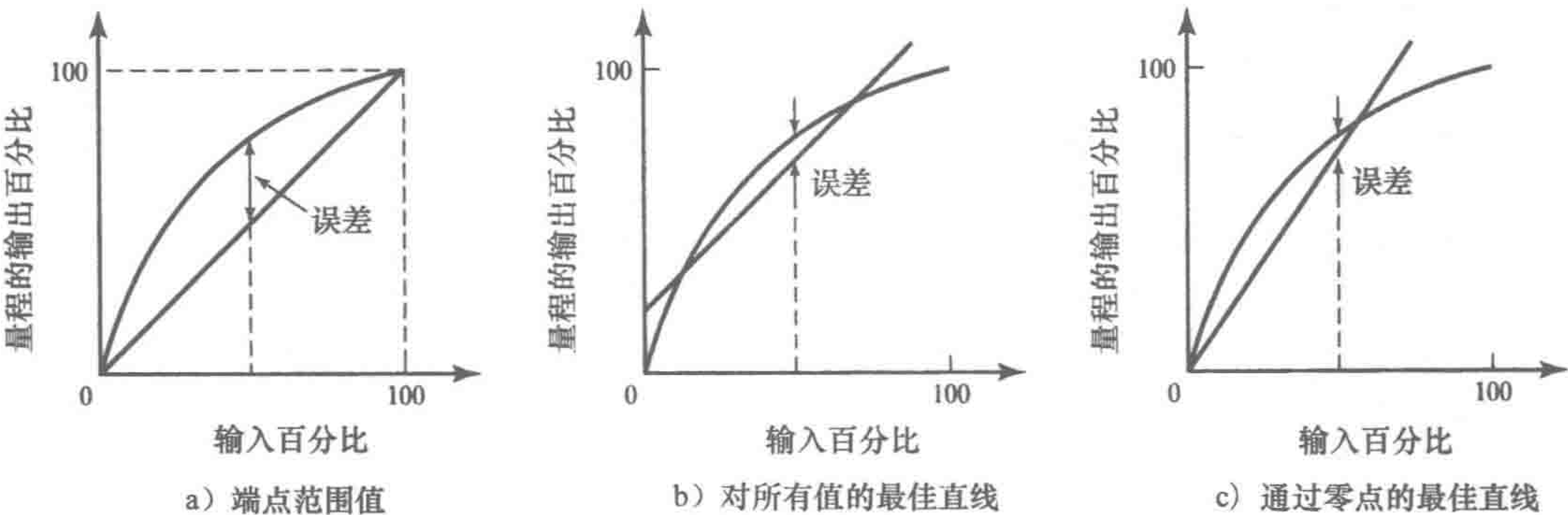


图 2-2　几种非线性误差定义方法

7. 可重复性/可再现性

一个变送器的可重复性（repeatability）和可再现性（reproducibility）是用于描述它在接收同一测量数据时给出相同输出的能力。重复性误差是由于反复测量同一个值但没有相同输出导致的，通常表示为整个测量范围的百分比。

31

$$可重复性=\frac{相同输入测得的最大值-最小值}{整个测量范围}\times 100$$

在一个特定的角速度下，一个用于测量角速度的传感器可以说有 ±0.01% 的测量范围的可重复性。

8. 稳定性

稳定性（stability）是变送器在一段时间内重复地测量一个常值给出相同输出的能力。**漂移**通常是用来描述输出随着时间的变化。漂移可以用测量范围的百分比来表示。**零点漂移**用来表示零输入时输出量的变化。

9. 死区和死区时间

变送器的死区（dead band）用来表示没有输出时的输入范围。例如转子流量计的轴承摩擦意味着只有流体达到临界速度阈值时才会有输出。死区时间（dead time）是从有输入到开始有输出响应的时间。

10. 分辨率

当输入在测量范围内连续变化时，有些传感器的输出信号可能会呈现小的阶跃变化。一个绕线电位计就是这样的传感器，当电位计滑块从一根线移到另一根线上时输出会阶跃变化。分辨率（resolution）是能够产生可观察输出量的最小输入变化值。对于一个绕线电位计，可以说它的分辨率是 0.5°或者是满刻度偏转的某一百分比。对于用数字输出的传感器，最小的输出信号是 1 位的。因此对于一个给出 N 位数据的传感器，总共有 2^N 位，分辨率就可以表示成 $1/2^N$ 。

32

11. 输出阻抗

当传感器的电输出与电路接口时，我们需要知道输出阻抗（output impedance），因为该阻抗与电路要么是并联的，要么是串联的。因此，传感器的输出阻抗能显著地改变所连接系统的性能。

为了说明以上几点，考虑应变式压力传感器的如下性能指标：

测量范围：70 ~ 1000kPa，2000 ~ 70 000kPa

电源电压：10V d. c 或者 a. c. r. m. s

满量程输出：40mV

非线性和迟滞误差：±0.5%满量程输出

温度范围：在 -54℃ ~ +120℃时工作

热零点漂移：0.030%满量程输出/℃

范围指标表示这个传感器可以测量 70 ~ 1000kPa 或者 20 ~ 70 000kPa 之间的压力。它在工作时需要 10V d. c 或者 a. c. r. m. s 供电。同时当低测量范围时达到 1000kPa 或者高测量范围达到 70 000kPa 时，给出输出值 40mV。非线性或者迟滞会导致在低测量范围内 1000 的 ±0.5% 的误差，即 ±5kPa 的误差；在高测量范围内 70 000 的 ±0.5% 的误差，即 ±350kPa 的误差。传感器能够在 -54℃ ~ +120℃温度范围内使用。当温度变化 1℃，传感器在 0 输入时输出会变化 0.030%，在低测量范围内 1000 的 0.030% = 0.3kPa，在高测量范围 70 000 的 0.030% = 21kPa 的零点漂移。

2.2.1　静态和动态特性

静态特性是当传感器满足稳定状态条件时的输出值，即在传感器接收到某个输入后，最终稳定下来的值。上面所定义的术语指的就是这样一个状态。**动态特性**指的是从传感器给出的输入值变化至达到稳态值的时间内的行为特性。动态性能是根据传感器针对某个输入的响应来描述的。例如，有一个从 0 突然变到某个值的阶跃输入，或者一个稳定速率的斜坡输入，或者某个固定频率的正弦输入。因此我们有以下术语（见第 19 章关于动态系统的更多的细节讨论）。

1. 响应时间

响应时间（response time）是在一个常值输入（阶跃输入）后，传感器的输出达到输入的某个比例所需要的时间，比如说 95%（见图 2-3）。例如一个水银温度计放到热水中，会有一个相当长的一个时间延迟，在温度计达到液体温度的 95% 时可能会有 100s 或者更长时间。

[33]

2. 时间常量

时间常量（time constant）是响应时间的 63.2%。一个在空气中的热电偶的时间常量为 40 ~ 100s。时间常量可以衡量传感器的惯性，同时也表示它对输入变化的响应快慢：时间常量越大，它对输入信号变化的响应速度越慢。我们将会在 19.3.4 节对阶跃输入下的系统的时间常量进行定量分析。

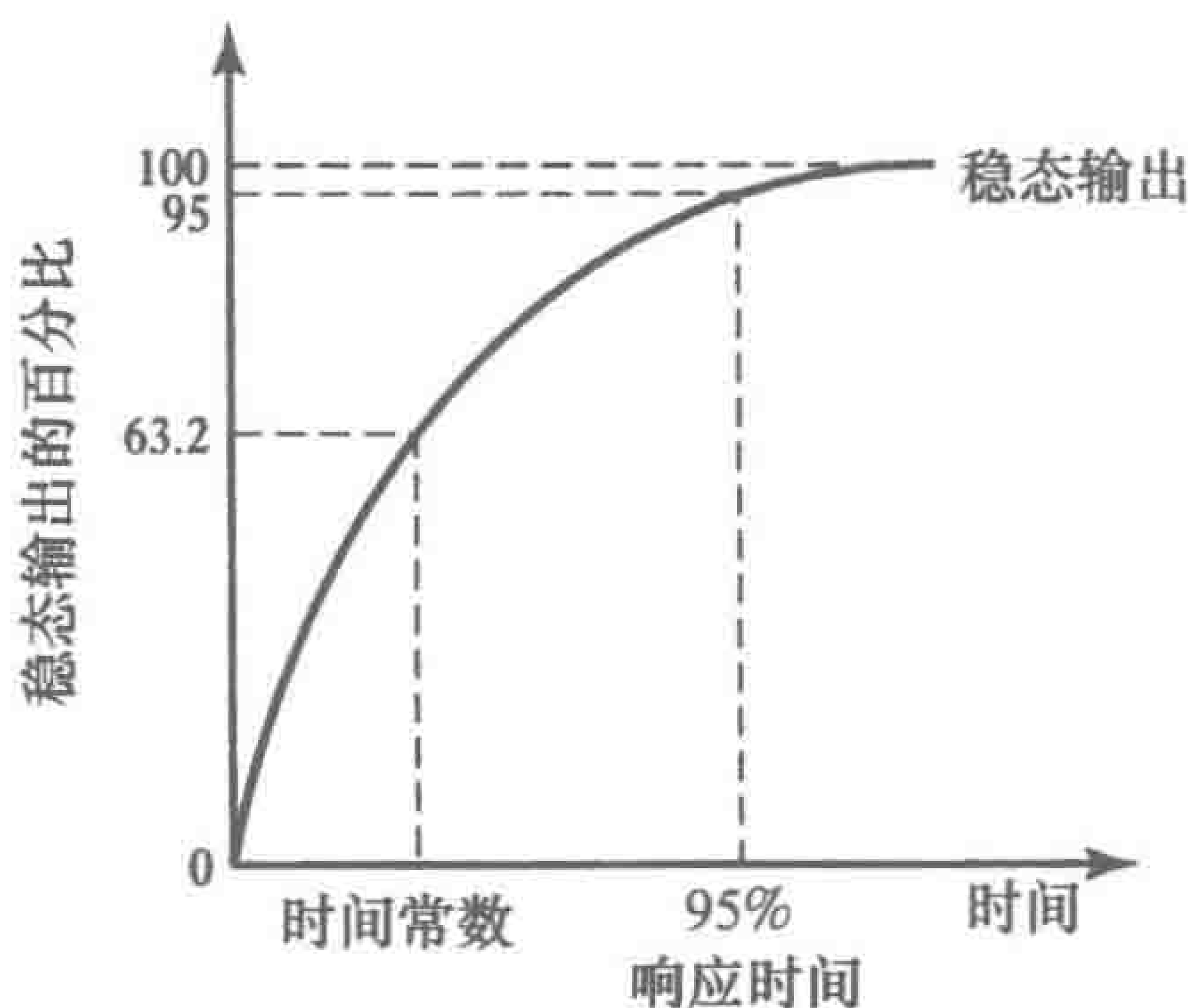


图 2-3　阶跃输入的响应

3. 上升时间

上升时间（rise time）是输出上升到稳态输出的某个百分比的时间。上升时间通常定义为响应曲线从稳态输出值的 10% 上升到稳态值 90% 或 95% 所需的时间。

4. 建立时间

建立时间（settling time）是输出达到稳态输出值的某个百分比之内，比如说 2% 之内所需的时间。

为了阐述以上几点，图 2-4 是一个在 $t=0$ 时插入液体中的温度计读数随时间变化的曲线。稳态值为 55℃，因为 55 的 95% 是 52.25℃，所以达到 95% 的响应时间为 228s。

[34]

根据它们所测量的量，下面的部分给出了传感器分组的一些例子。这些被测量都是在实际机械工程中经常遇到的，比如说：位移、接近、速度、力、压强、流量、液位、温度和光强。

2.3　位移、位置和接近

位移传感器关心的是物体移动了多少；**位置**传感器关心的是物体对于某个参考点的相对位置；**接近**传感器是一种位置传感器，它用于确定物体是否处于传感器的某个特殊的临界距离内。它们基本上是给出开/关输出的设备。

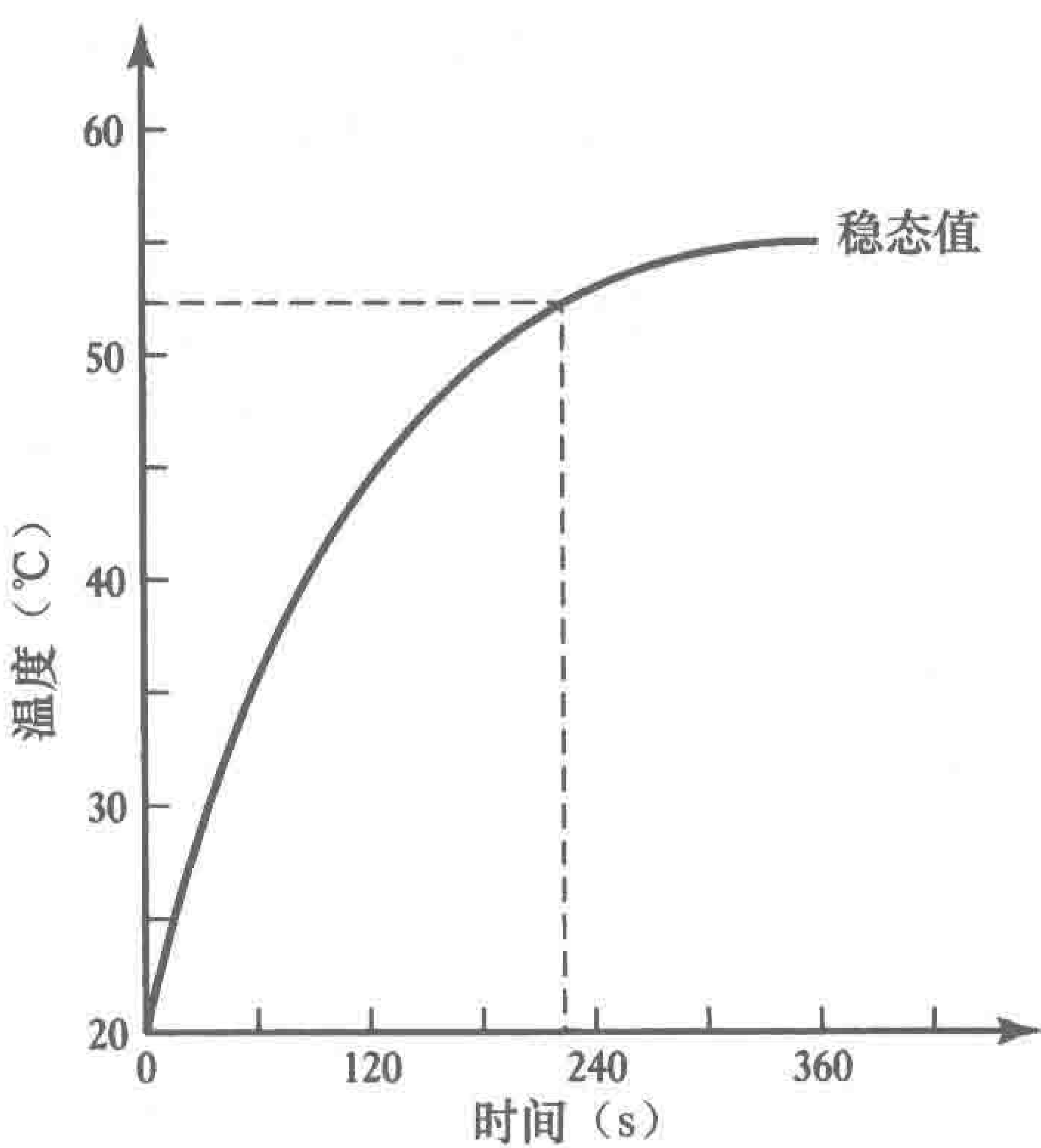


图 2-4　液体中的温度计

位移和位置传感器分为两种基本类型：一种是被测物体和传感器直接接触，另一种是被测物体与传感器没有直接接触。对于接触式线性位移测量方法，通常有一个和被测物体直接接触的传感轴。这个轴的位移通常通过传感器监测。轴的移动会导致其电压、电阻、电容或者互感的变化。对于接触式测量角位移的方法，轴的转动可以通过齿轮直接驱动传感器元件转动。非接触传感器和被测物体附近会存在与导致传感器气压的变化，或者电感电容的变化的有关因素。下面是常用的位移传感器。

2.3.1　电位器传感器

一个**电位器**是由一个具有滑动触点的电阻元件组成，其中滑动触点能够与电阻元件滑动接触。这样的元件能够用于测量线性或者旋转位移，位移能转化为电势差。旋转电位器由一个圆的绕线轨道或者一个上面有旋转滑块的导电塑料薄膜组成（见图 2-5）。轨道可以是单圈的或者多圈的（螺旋的）。在 1、3 之间有固定的输入电压 V_s，在 2、3 之间的输出电压 V_o 是输入电压的一部分，其大小取决于在 2、3 之间的电阻 R_{23} 与 1、3 之间的总电阻 R_{13} 的比值，即 $V_o/V_s = R_{23}/R_{13}$。如果轨道在单位长度上的电阻值是固定的，也就是在单位角度上的电阻是定值，那么输出电压就与滑块旋转的角度成正比。因此角位移就可以转化为电势差。

35

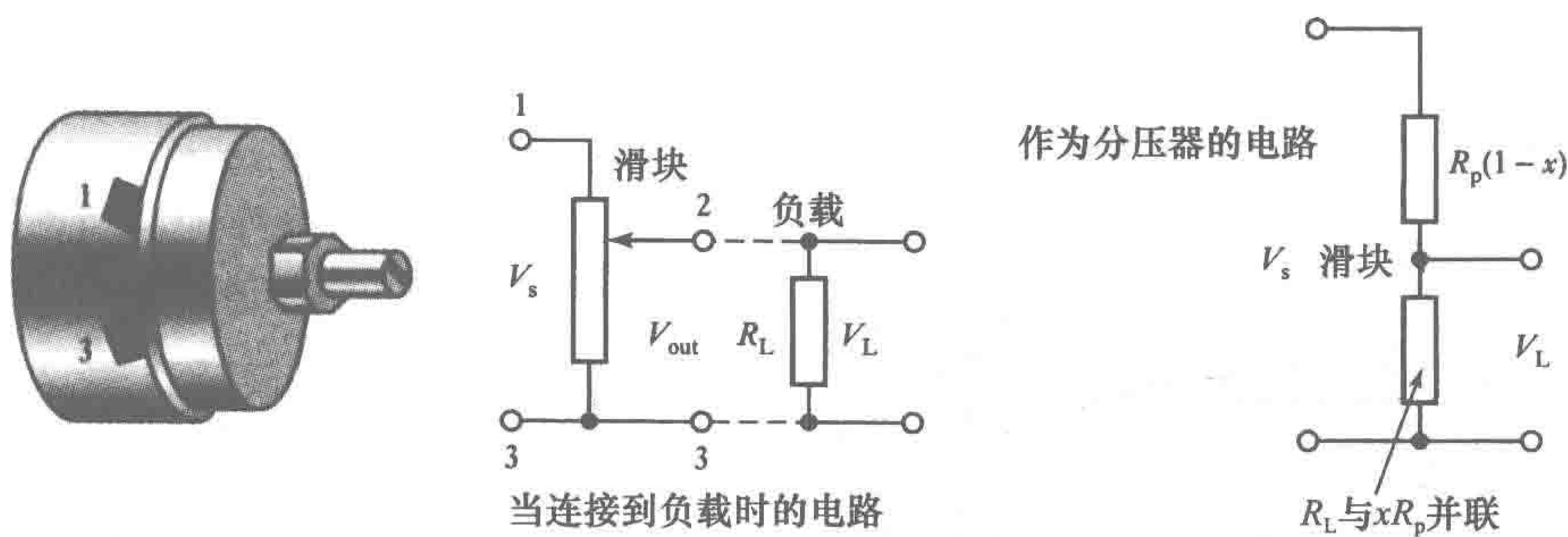


图 2-5　旋转电位器

对于绕线的轨道，当一个滑块沿着绕线轨道从一圈到另一圈会导致输出电压呈步进变化。每一圈就是一步。如果电位计有 N 圈，那么分辨率按百分比计算就是 $100/N$。这样，一个绕线轨道的分辨率受使用的线的直径限制，通常 1.5mm 线的直径用于不精确的绕线轨道，0.5mm 线的直径用于精确的绕线轨道。由轨道的非线性导致的误差从小于 0.1% 到大约 1% 之间。轨道电阻在 20Ω ~ 200kΩ 之间。导电塑料有着理想的无限分辨率，由于轨道的非线性会导致 0.05% 数量级的误差，电阻值在 500Ω ~ 80kΩ 之间。导电塑料比电线有更高的电阻温度系数，因此温度变化对精度有更大的影响。

一个对电位器有重要影响的因素是连接输出的负载电阻 R_L。只有当负载电阻为无穷大时，负载上的电势差 V_L 才和输出电压 V_o 成正比。然而，对于有限负载，负载效应使角度和输出电压的线性关系转化成非线性关系。负载电阻 R_L 是与电位器电阻 R_p 的 x 部分并联的。并联电阻为 $R_L x R_p/(R_L + xR_p)$。电源电压上的总电阻为

$$\text{总电阻} = R_p(1 - x) + R_L x R_p/(R_L + xR_p)$$

该电路是一个分压器电路，因此负载电压和总电压的比值是负载上电阻与总电阻的比值：

$$\frac{V_L}{V_s} = \frac{xR_L R_p/(R_L + xR_p)}{R_p(1 - x) + xR_L R_p/(R_L + xR_p)}$$

$$= \frac{x}{(R_p/R_L)(1 - x)x + 1}$$

如果负载是无穷电阻，则有 $V_L = xV_s$。于是，因负载有着有限的电阻而导致的误差为：

$$\text{误差} = xV_s - V_L = xV_s - \frac{xV_s}{(R_p/R_L)x(1 - x) + 1}$$

$$= V_s \frac{R_p}{R_L}(x^2 - x^3) \Big/ \left[\frac{R_p}{R_L}x(1 - x) + 1\right] \quad \text{当}\frac{R_p}{R_L}x(1 - x) \ll 1 \text{ 时，分母约等于 } 1\text{。}$$

为了说明以上几点，考虑一个 500Ω 电阻的非线性误差，当滑块位移达到最大距离的一半时，且负载电阻为 10kΩ。电源电压为 4V。利用上面的推导，有

36

$$\text{误差} = 4 \times \frac{500}{10\,000}(0.5^2 - 0.5^3) = 0.025\text{V}$$

作为满量程读数的百分比是 0.625%。

电位器可用作汽车电子系统中的加速器踏板和节流阀位置检测的传感器。

2.3.2　应变片

电阻应变片（见图 2-6）可以是金属丝、金属箔条，或者一条晶片状的半导体材料，并且要能够像邮票一样粘在表面上。当有应变产生时，它的电阻 R 发生变化，电阻变化 $\Delta R/R$ 与应变 ε 成正比，即

$$\frac{\Delta R}{R} = G\varepsilon$$

G 是一个比例系数，称为应变系数。

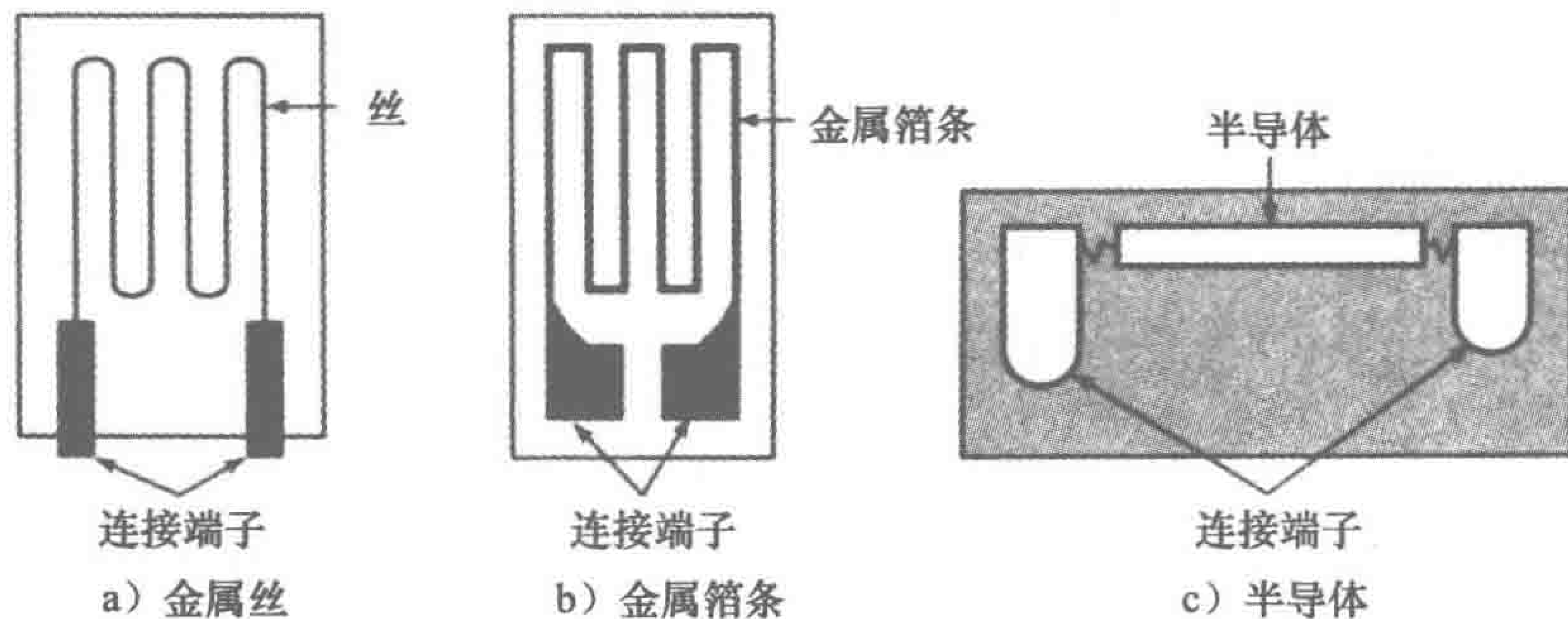


图 2-6　应变片

由于应变是一个比值（长度的变化/原长），所以应变片的电阻变化测量的是应变片所黏附的元件长度的变化。金属丝或者箔式应变片的应变系数一般是 2.0，其电阻的数量级一般为 100Ω。对于 p 型、n 型硅半导体，p 型半导体的应变片灵敏系数为 +100 或者以上，n 型半导体的应变片灵敏系数为 -100 或者更多，电阻的数量级为 1000～5000Ω。应变系数通常是在应变

片制造时从一批元件中选取一些样品校正得到的。校正需要使应变片样品产生已知的应变，然后测量它们电阻的变化。所有的应变片都存在一个问题，就是它们的电阻不光会随着应变变化，而且会随着温度变化。消除温度影响的方法在第 3 章进行讨论。半导体应变片比金属应变片对温度的敏感性更高。

为了说明以上几点，考虑一个电阻为 100Ω，应变系数为 2.0 的应变片。当它受到 0.001 的应变时，应变片的电阻变化是多少？电阻的变化等于应变系数乘以应变，因此

$$\text{电阻的变化} = 2.0 \times 0.001 \times 100 = 0.2\Omega$$

37

一种位移传感器是把应变片粘在悬臂（见图 2-7a）、环（见图 2-7b）、U 型（见图 2-7c）等复杂形状的结构上。当由于接触点位移产生力，进而导致这些柔性的元件弯曲或者变形时，粘在元件上的电阻应变片受到应变并给出了一个能够检测到的电阻变化。因此电阻变化可以度量元件的位移或者变形。这种元件通常用于数量级为 1 ~ 30mm 或者非线性误差为满量程 ±1% 的线性位移的测量。

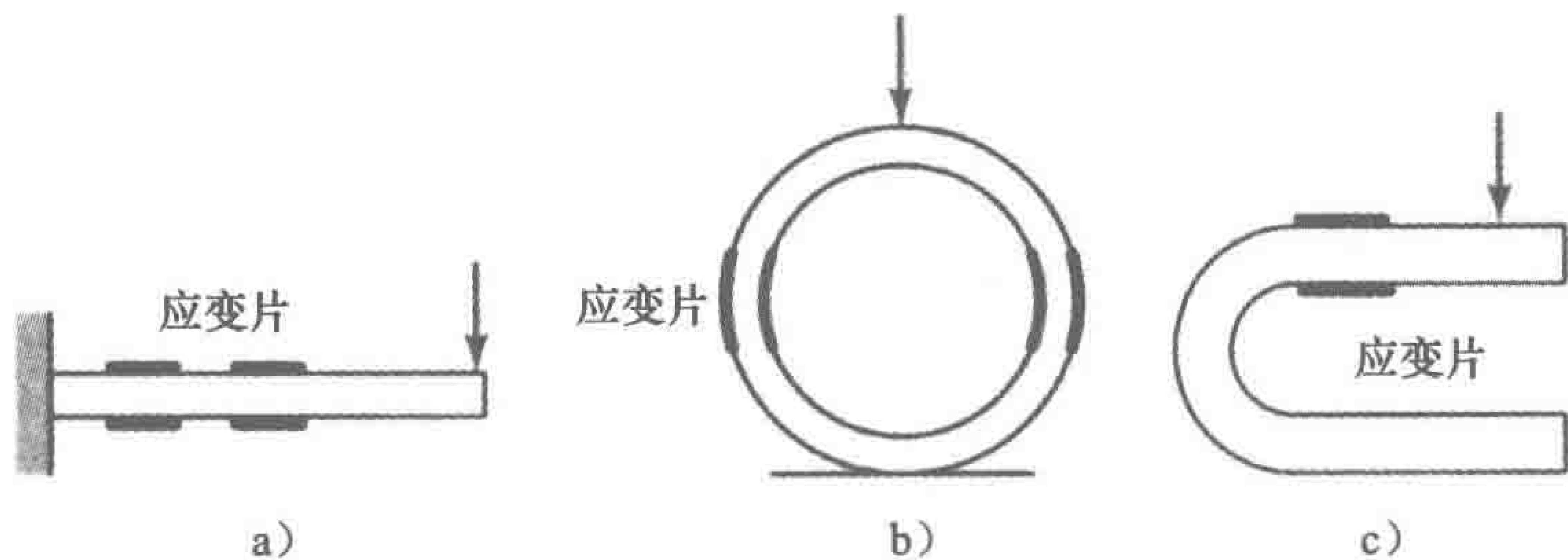


图 2-7　应变元件

2.3.3　电容元件

一个平行板电容器的电容值为

$$C = \frac{\varepsilon_r \varepsilon_0 A}{d}$$

ε_r 是两板之间电介质的相对介电常量（或电容率），ε_0 是真空的电容率，A 是两板之间的重叠面积，d 是两板之间的距离。于是，电容式传感器检测线性位移可采用图 2-8 的形式。图 2-8a 是位移导致两板之间的距离发生变化；图 2-8b 是位移导致两板之间的重叠面积发生变化；图 2-8c是位移导致两板之间的电介质发生变化。

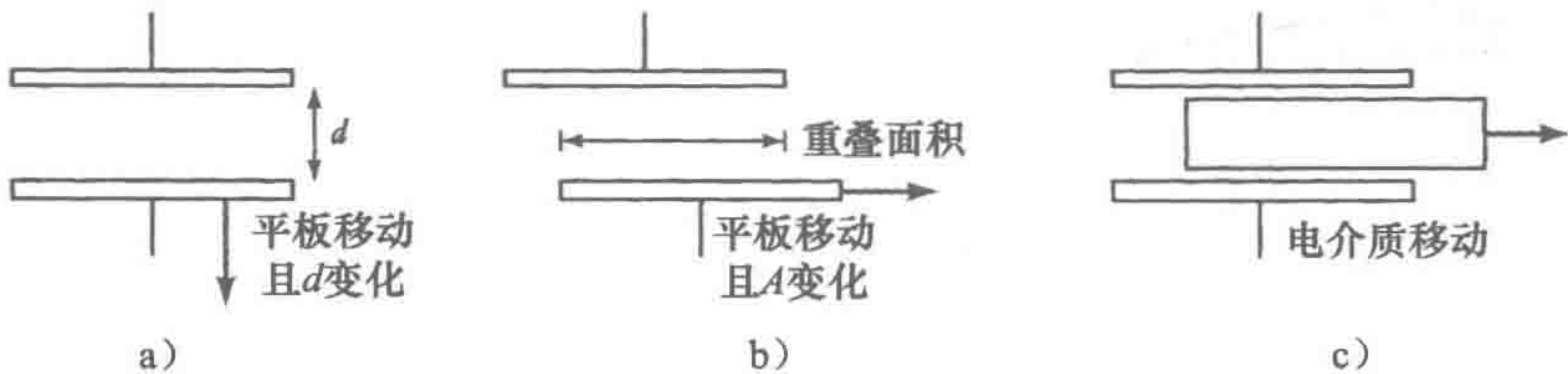


图 2-8　电容式传感元件的形式

对于板间距变化的位移，如果间距 d 增加了位移 x，那么电容的变化为

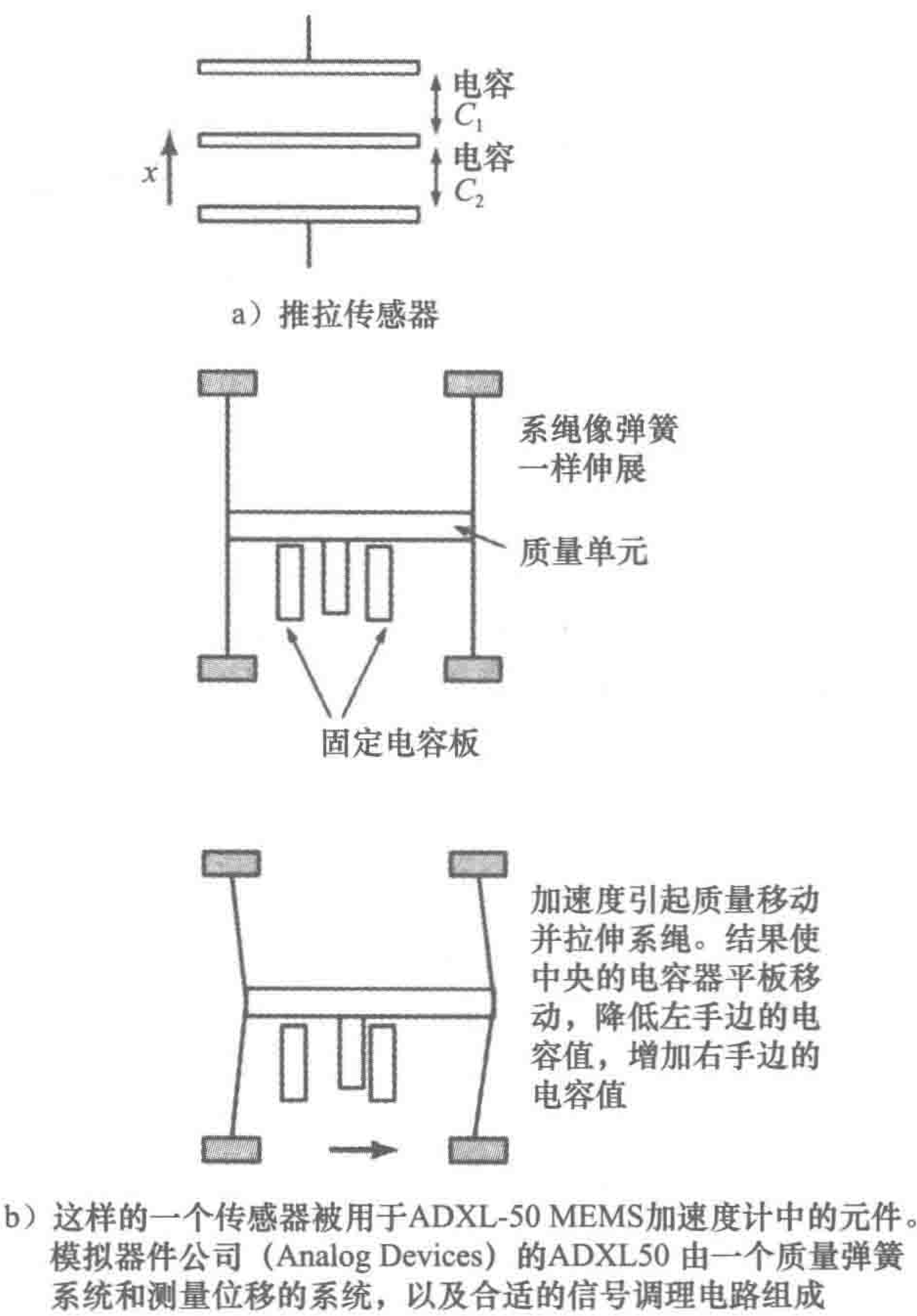
$$C + \Delta C = \frac{\varepsilon_0 \varepsilon_r A}{d + x}$$

因此电容变化量 ΔC 和初始电容 C 的比值是：

38

$$\frac{\Delta C}{C} = \frac{d}{d+x} - 1 = -\frac{x/d}{1+(x/d)}$$

因此电容的变化值 ΔC 和位移 x 呈非线性关系。这种非线性关系能够通过使用一种叫作**推拉位移传感器**（见图 2-9a）克服。图 2-9b 显示它如何在实际中实现。有三个平板，上面两个形成一个电容，下面两个形成另一个电容，位移移动中间那块板子，结果是中间板子向下移动增加了上面电容的间距，减小了下面电容的间距。因此，有



a）推拉传感器

b）这样的一个传感器被用于ADXL-50 MEMS加速度计中的元件。模拟器件公司（Analog Devices）的ADXL50 由一个质量弹簧系统和测量位移的系统，以及合适的信号调理电路组成

图 2-9　推拉位移传感器原理

$$C_1 = \frac{\varepsilon_0 \varepsilon_r A}{d+x}$$

$$C_2 = \frac{\varepsilon_0 \varepsilon_r A}{d-x}$$

当 C_1 是一个交流电桥的臂，C_2 是另一个时，这样导致的不平衡电压与 x 成正比。这样的传感器通常用于检测几毫米到几百毫米的位移。非线性和迟滞误差通常为满量程的 ±0.01%。

一种接近式电容传感器是由单个电容平面探针通过接近另一个由金属材料制成并接地的物体表面组成的（见图 2-10）。随着物体接近，电容的平板间距发生变化，当物体离探针很近时，电容变化很明显并且可以探测到。

39

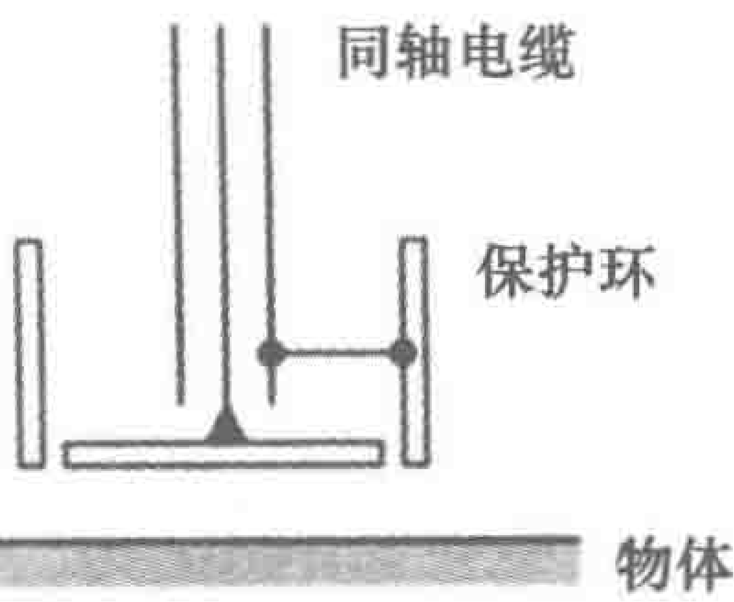


图 2-10　接近电容传感器

2.3.4　差动变压器

线性可变差动变压器（LVDT），由三个对称安装在绝缘管上的线圈组成（见图 2-11）。中间的线圈是初级线圈；另两个是一样的反向串联的次级线圈，因此它们的输出是相反的。一个需被检测的位移会使磁心在中间的管中发生移动。

当有一个变化的电压输入到初级线圈时，会在次级线圈产生变化的感应电动势。当磁心在中间时，因为在次级线圈中的磁性材料是一样的，所以在每个次级线圈感应的电动势是相同的。又由于它们的连接方式使它们的输出相反，所以净输出是零。

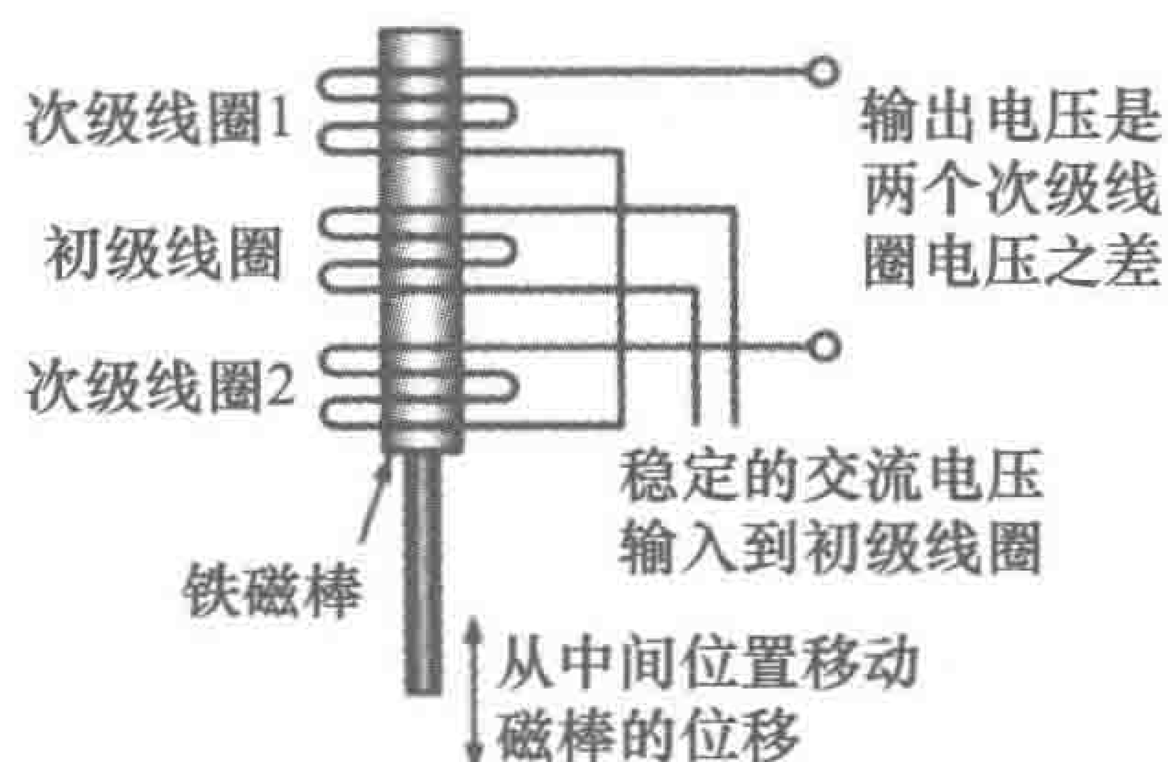


图 2-11　LVDT

然而，当磁心偏离中间位置时，一个线圈中的磁心比另一个大，比如 2 线圈中的磁心比 1 线圈中的大。结果是一个线圈感应的电动势比另一个大。那么两个线圈会有一个净输出。因此一个更大的位移就意味着一个线圈中的磁心比另一个中的更多，那么被测量的位移越大，两个线圈的感应电动势之差形成的输出也就越大（见图 2-12）。

由于初级线圈的电流变化导致次级线圈中感应的电动势为

$$e = M\frac{\mathrm{d}i}{\mathrm{d}t}$$

M 为互感系数，它的值取决于线圈的匝数和铁磁心。因此，对于一个初级线圈的正弦输入电流 $i = I\sin\omega t$，1 和 2 两个次级线圈感应的电动势为

$$v_1 = k_1\sin(\omega t - \phi) \text{ 和 } v_2 = k_2\sin(\omega t - \phi)$$

k_1、k_2 和 ϕ 取决于在某个特殊磁心位置时初级和次级线圈的耦合程度。ϕ 是初级交变电压和次级交变电压之间的相位差。由于两个输出是串联的，所以它们的差值是：

$$\text{输出电压} = v_1 - v_2 = (k_1 - k_2)\sin(\omega t - \phi)$$

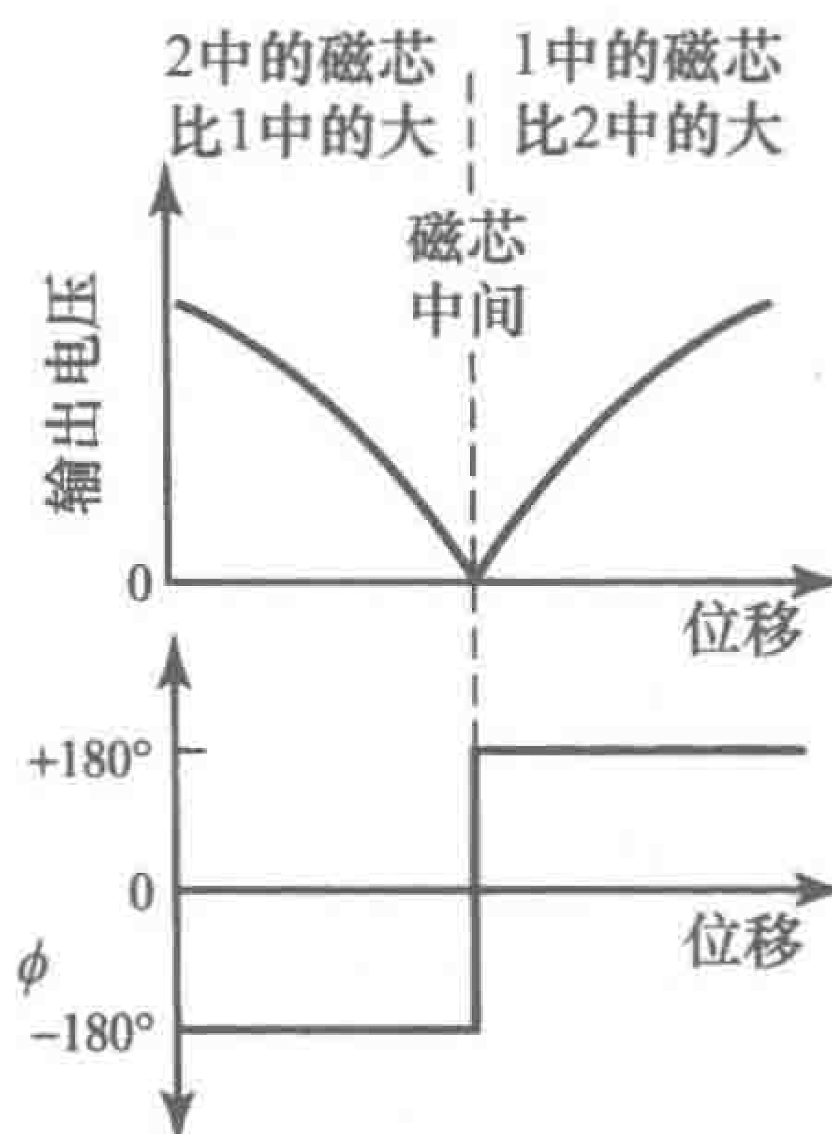


图 2-12　LVDT 输出

当在两个线圈中的磁心都是一样的时，k_1 和 k_2 相等，则输出电压是零。如果 1 中的磁心比 2 中的大，则 $k_1 > k_2$，有：

$$\text{输出电压} = (k_1 - k_2)\sin(\omega t - \phi)$$

当 2 中的磁心比 1 中的大时，有 $k_1 < k_2$。当磁心从在 1 中更多向在 2 中更多的方向移动时，会有一个 180°的相角变化。因此

$$\text{输出电压} = -(k_1 - k_2)\sin(\omega t - \phi) = (k_2 - k_1)\sin[\omega t + (\pi - \phi)]$$

图 2-12 显示的是输出电压的大小和相角随着磁心的位移变化的曲线。

用这种输出形式，两个不同的位移会产生相同大小的输出电压。为了使每个不同的位移有不同的输出电压，我们必须区分那些大小相同但相角差为 180°的情况。一个有着低通滤波的相敏解调器（phase-sensitive demodulator），用于把输出转化为与每个位移唯一对应的直流电压值（见图 2-13）。这样的电路已经有现成的集成电路。

通常线性可变差动变压器的工作范围为 ±2 ~ ±400mm，非线性误差大约为 ±0.25%。LVDT 作为初级变送器广泛用于监测位移。磁心的自由端可以和被检测的表面以弹簧形式接触，也可以通过螺纹形式机械连接。LVDT 也可以作为次级变送器用于测量力、重量和压力。将这些物理量转换为位移量，这样就能被 LVDT 检测到了。

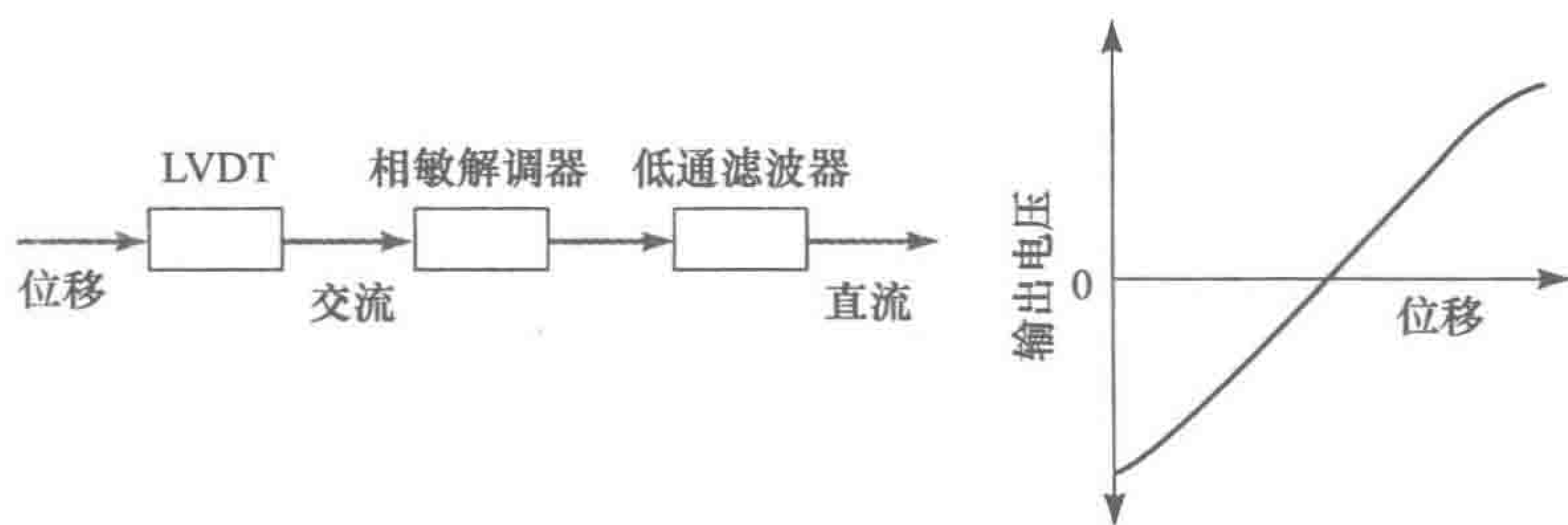


图 2-13　LVDT 直流（d. c.）输出

一个旋转可变差动变压器（RVDT）能够用于测量旋转（见图 2-14），它与线性可变差动变压器的原理一样。磁心是心形曲线的磁性材料，旋转能使通过一个二次绕组的磁心比通过另一个二次绕组的磁心多。典型的测量范围是 ±40°，线性误差约是测量范围的 ±0.5%。

2.3.5　涡流式接近传感器

如果线圈通过交变的电流，就会产生一个交变的磁场。如果有一个金属物体在这个交变的磁场附近，那么金属内部会产生涡流。涡流本身会产生一个磁场，这将使产生它们的原来的磁场发生扭曲。结果是线圈的阻抗发生变化，从而使交变电流的幅值也发生变化。在某些预设的等级值，这些变化能够触发一个开关。图 2-15 是一个这种传感器的基本形式，它用于检测非磁性但是导电的材料。它们有价格相对便宜、体积小、可靠性高，同时对小位移有高敏感性等优点。

41

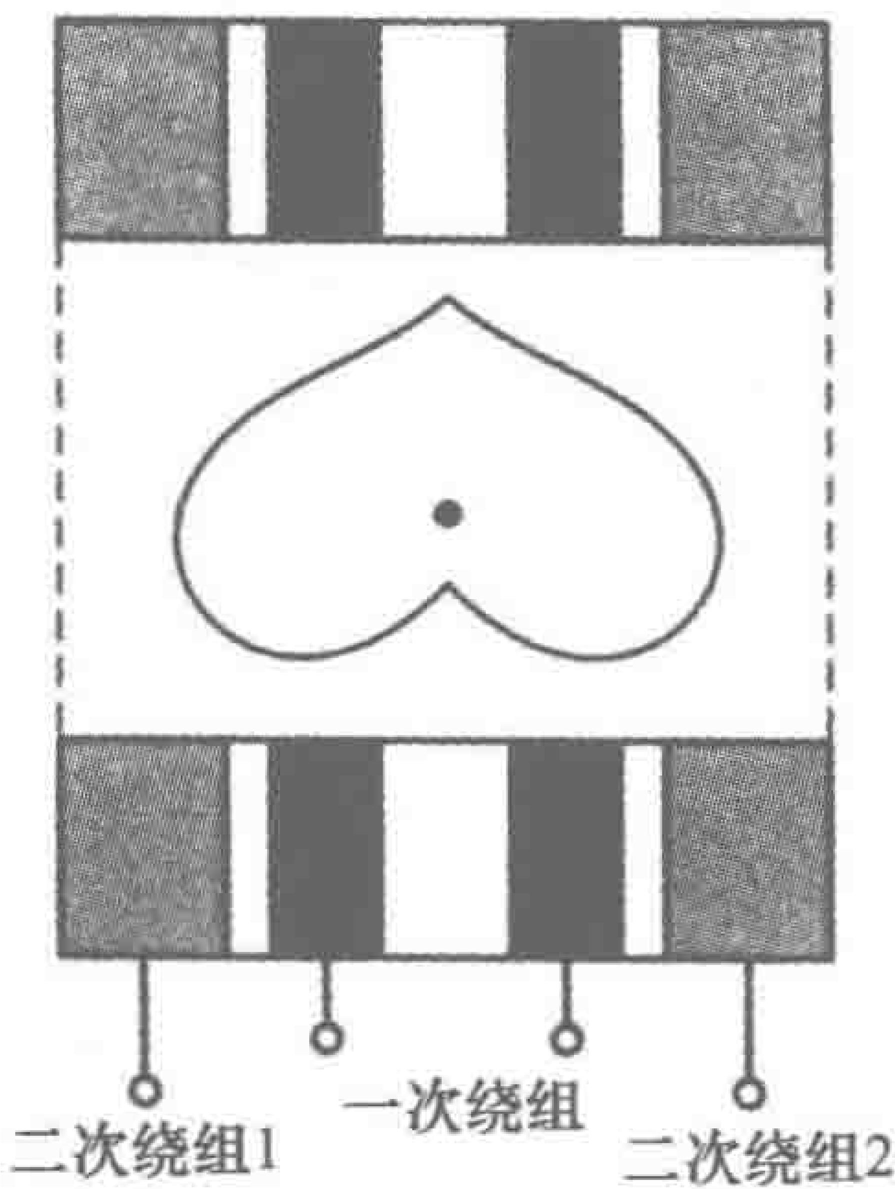


图 2-14　RVDT

2.3.6　电感式接近开关

电感式接近开关由一个缠绕磁心的线圈组成。当线圈的末端接近金属物体时，它的电感会发生变化。这个变化能够通过它对谐振电路产生的影响来检测，然后这个变化用于触发一个开关。这个只能用于检测金属物体，并且最好能够用于铁金属。

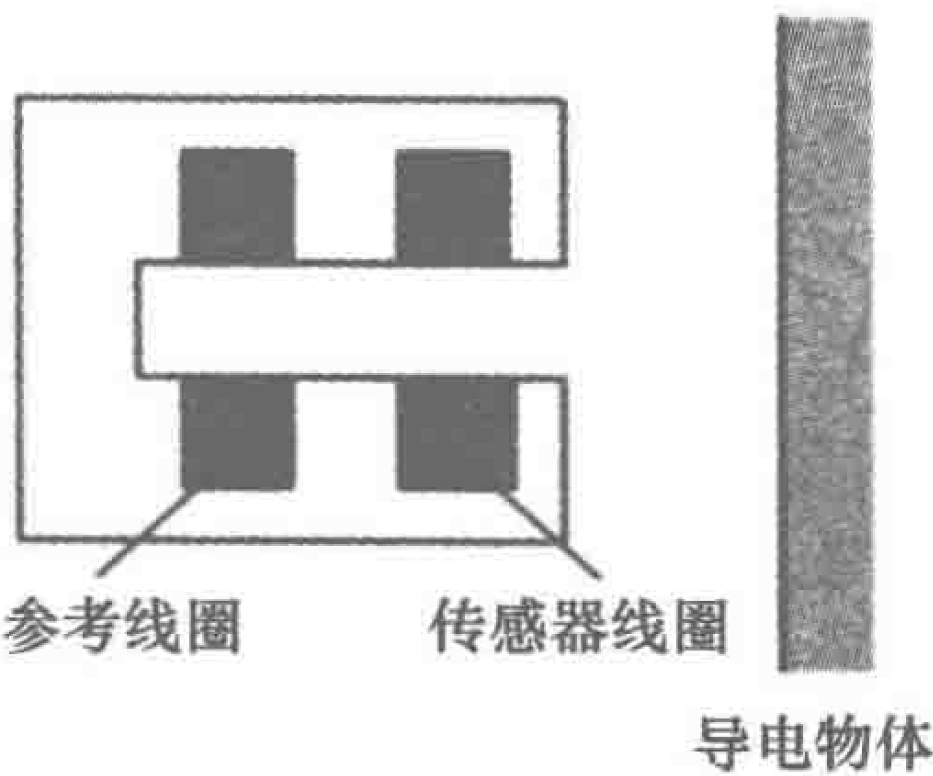


图 2-15　涡流式传感器

2.3.7　光学编码器

编码器是能够对线性或者角位移输出数字信号的装置。位置编码器可分为两类：能够检测相对于某个基准的旋转变化的**增量编码器**和能够给出实际的角度位置的**绝对编码器**。

图 2-16a 显示的是角位移增量式编码器的基本形式。通过码盘狭缝的一束光，能够被合适的光学传感器检测到，当码盘转动时，传感器会产生脉冲输出，脉冲的数量与码盘转过的角度成正比。因此码盘和旋转码盘轴的角位置，能够由某个基准位置开始产生的脉冲数量决定。在实际应用中使用了三个有传感器的同心轨道（见图 2-16b）。最里面的轨道有一个孔用于定位码盘的“零”位。另外两个轨道分别有一系列等间距分布的孔布满整个圆周。但是中间轨道偏移外圈轨道的孔有半个孔的宽度。这个偏移保证旋转的方向能够被检测到。当顺时针旋转时，外圈轨道对应的脉冲超前中间轨道的脉冲 90°；当逆时针旋转时，外圈轨

道对应的脉冲滞后于中间轨道脉冲 90°。编码器的分辨率由码盘轨道的缝隙数量来决定。由于一圈是 360°，所以如果一圈有 60 个缝隙，则分辨率为 360/60 = 6°。

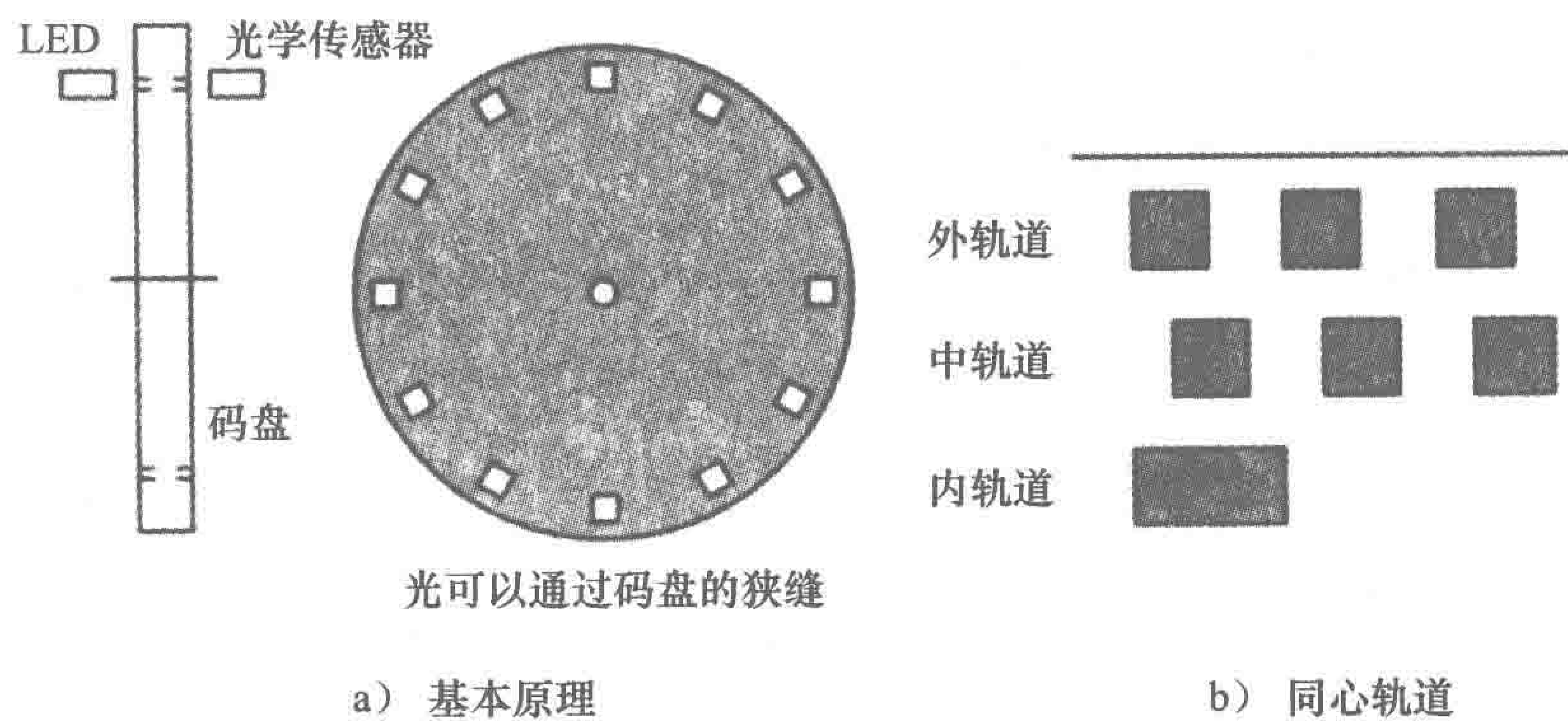


图 2-16　增量式编码器

图 2-17 显示的是角度位移测量的绝对编码器的基本形式。它给出了二进制数字形式的输出量，每一个这样的数字表示一个特殊的角度位置。旋转盘有三个带槽的同心圆和三个传感器检测光脉冲。这样的安排使传感器的串行输出量是二进制表示的数字。传统的编码器最高有 10 ~ 12 个轨道。二进制的位数等于轨道的数量。因此 10 个轨道就会有 10 位，所以位置的数量能被检测到的就是 2^{10}，也就是 1024 个，分辨率为 360/1024 = 0.35°。

我们通常不用正常形式的二进制码，因为从一个二进制数变到下一个二进制数可能导致不止一位的变化，比如从 101 到 110 有两位发生变化，如果因为错位导致某一位在其他位之前发生了部分的变化，那么会瞬间产生一个中间的二进制数，导致错误计数。为了克服这一点，我们使用**格雷码**（见附表 B）。这种编码方法从一个数变到下一个数时只有 1 位发生变化。图 2-18显示的是正常编码和格雷编码的轨道安排。

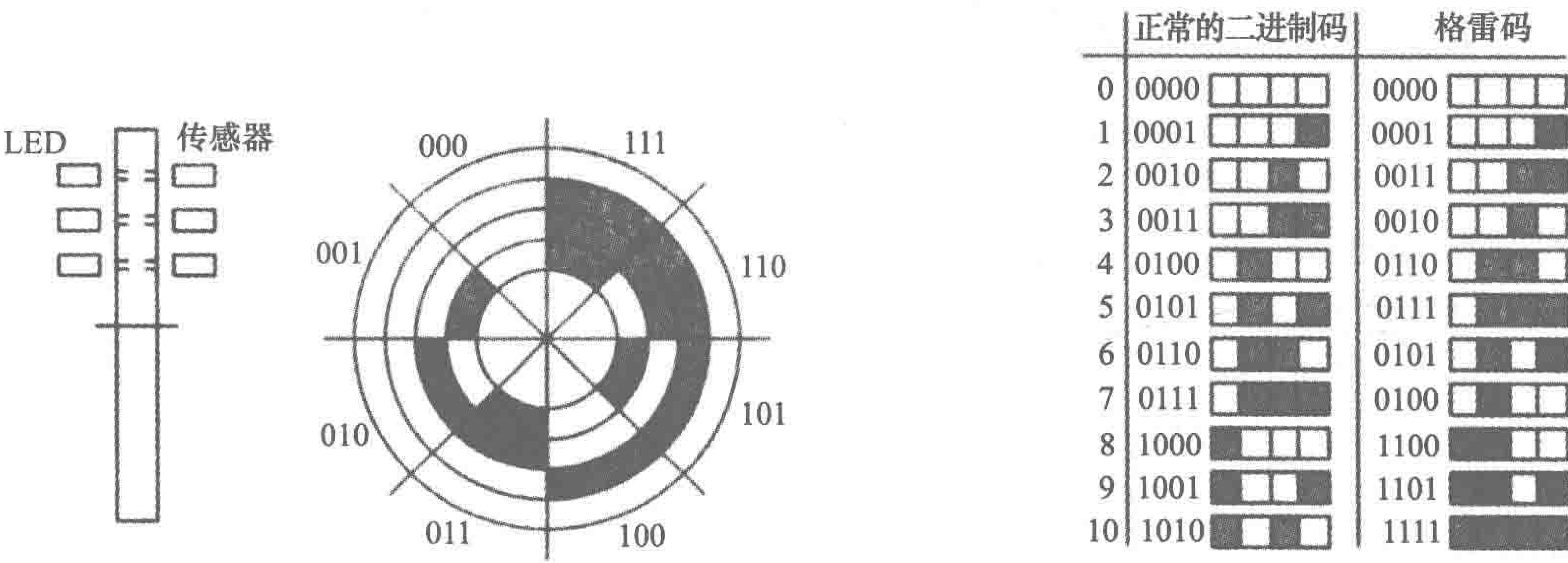


图 2-17　一个 3 位的绝对编码器　　图 2-18　二进制码和格雷码

光学编码器，比如说惠普的 HEDS-5000，包括安装在轴上的一个 LED 灯和一个编码盘。同时也提供了集成电路接口来解码，并把格雷码转换成适合微处理器的二进制输出形式。对于一个在码盘上有 7 个轨道的绝对编码器，每个轨道在二进制位上表示 1 位，这样我们有 2^7 个不同的位置，也就是 128 个，8 个轨道就是 2^8 个不同的位置，也就是 256 个。

2.3.8　气动传感器

43

气动传感器使用了压缩空气，物体的位移或者接近能够转换为气压的变化。图 2-19 表示了这样一种传感器的基本形式。低压空气能够通过传感器前面的出口逸出。在附近没有物体的情况下，逸出的空气会使传感器输出口附近的压力减小。然而，如果附近有物体，空气就不能很容易地逸出，从而导致传感器输出口的压力增加。输出压力取决于物体的接近程度。

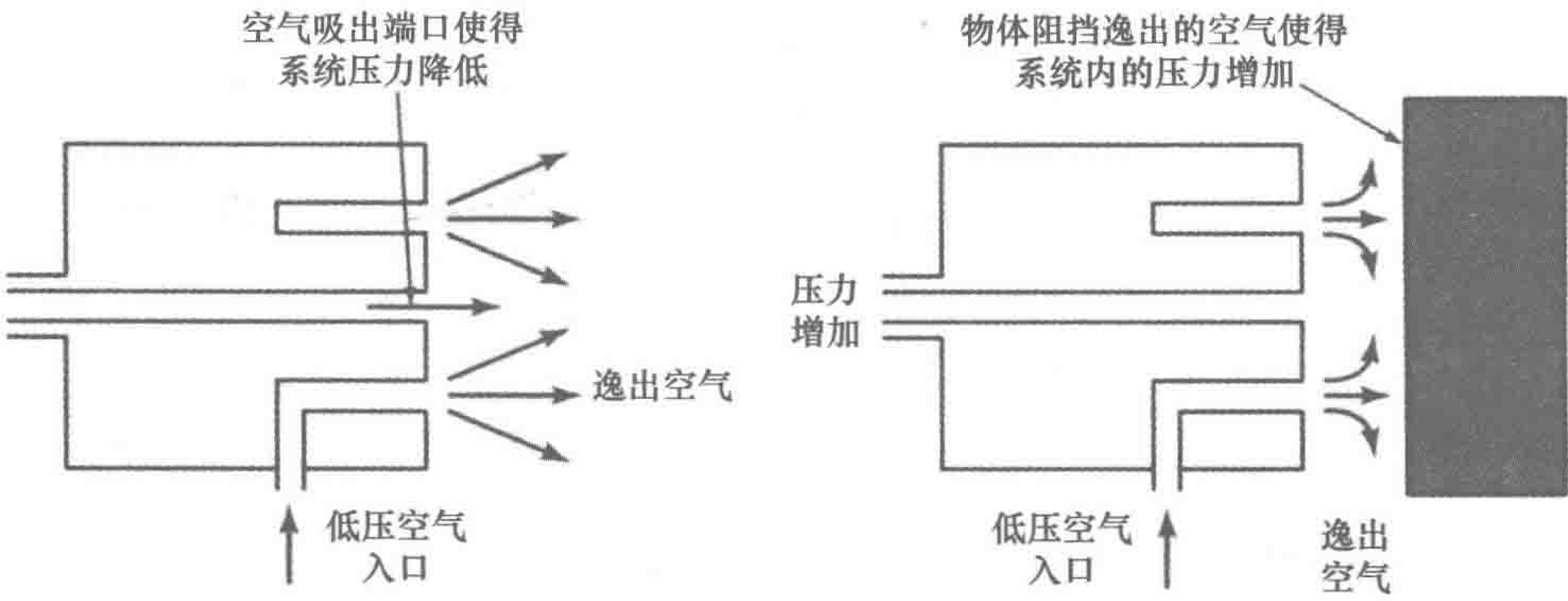


图 2-19　气动接近传感器

这样的传感器一般用于测量在 3 ~ 12mm 范围内的位移变化。

2.3.9　接近开关

有几种形式的开关能够被一个物体的存在而激活，从而使接近传感器给出开/关的输出量。

微动开关是一个小的电气开关，它需要物理接触，一个很小的操作力就能使其接通。例如，为了判别传送带上是否有物体存在，通过物体的重量压着传送带，然后带动下面的弹簧承载平台，这样平台的移动就合上了开关。图 2-20 给出了几种能够触发开关的例子。

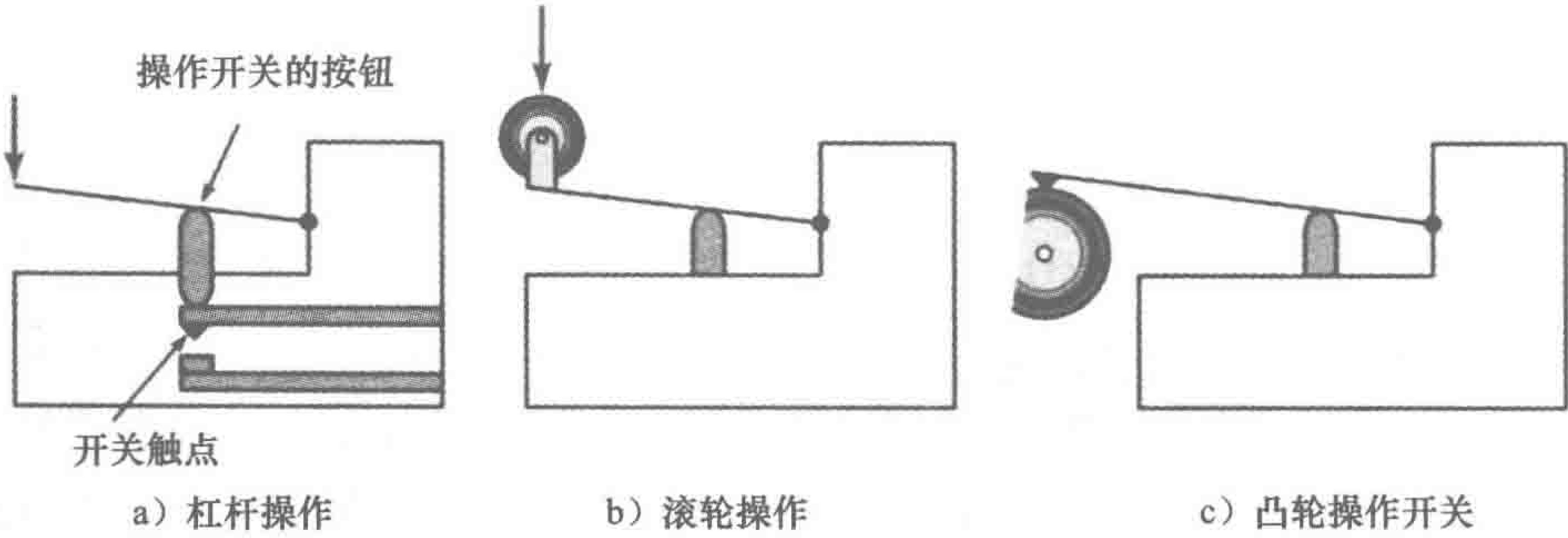


图 2-20　几种能够触发开关的例子

图 2-21 显示的是**簧片开关**的基本形式。它由密封在玻璃管内的两个磁性触点组成。当一个磁铁接近开关时，两个磁性簧片开关彼此吸引从而闭合开关。这是一个非接触式的接近开关。这种开关广泛应用在检测门是否合上。它也用于类似转速计的装置，这种装置由带齿的轮子旋转使簧片开或关。如果其中一个齿贴上磁铁，这样每次它通过开关就会暂时合上开关，然后在相关的电路中产生一个电流/电压脉冲。

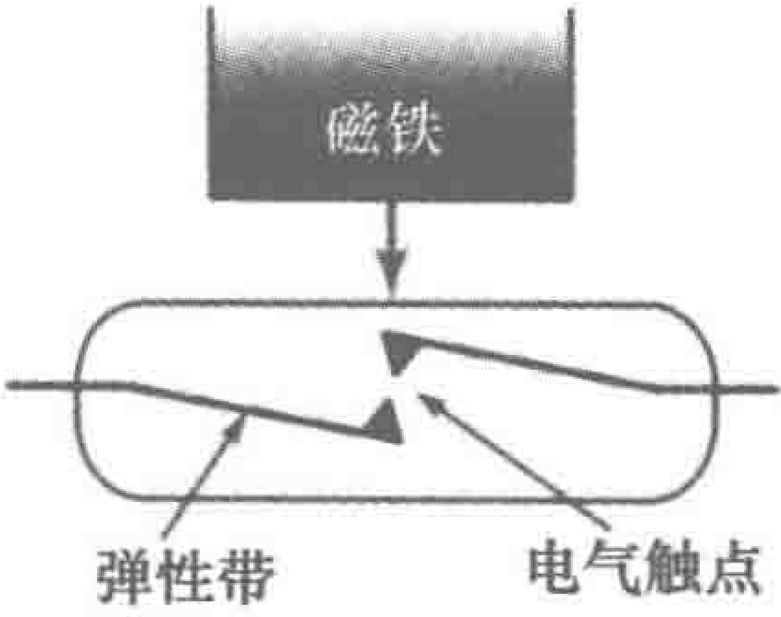


图 2-21　簧片开关

光敏装置是通过阻挡光束，或者红外光照射到这个装置，来检测一个不透明物体是否存在或者通过检测被物体反射的光（见

图 2-22)。

44

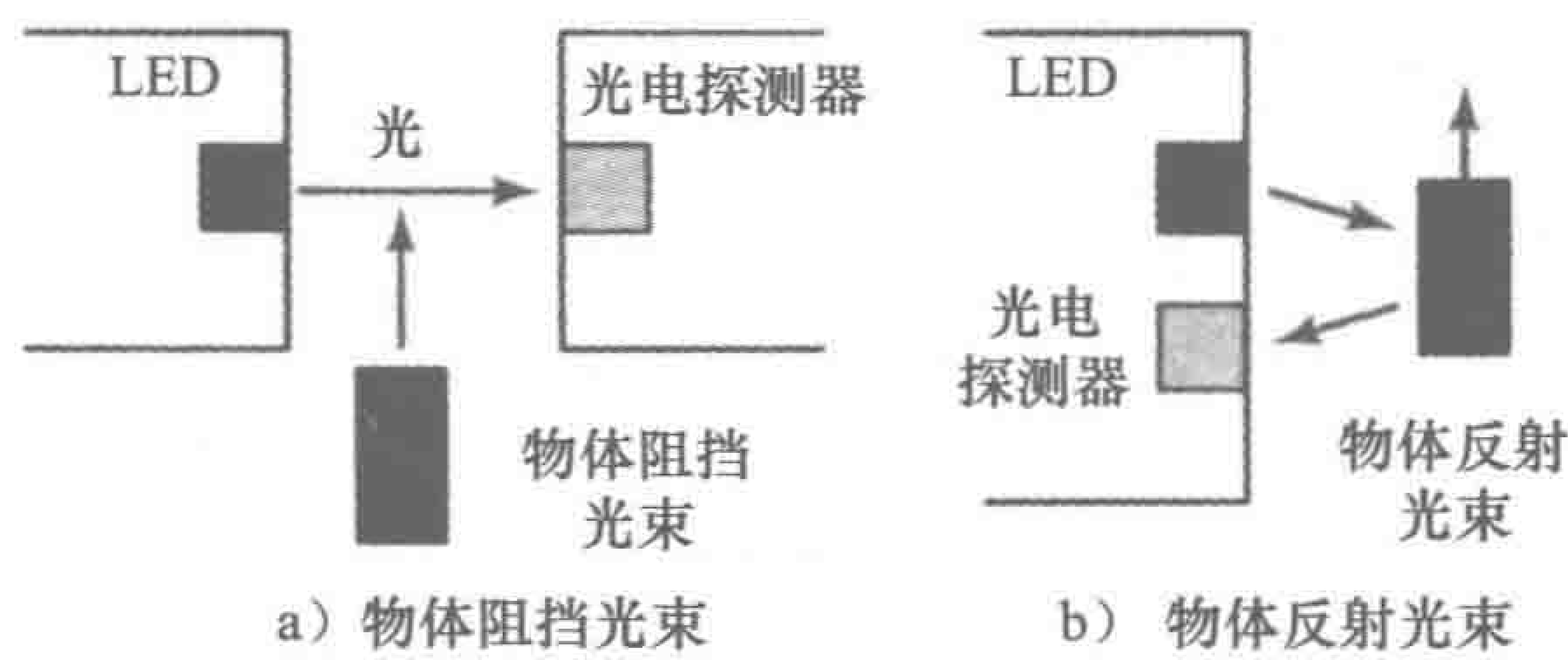


图 2-22 用光电传感器来检测物体

2.3.10 霍尔效应传感器

当一束带电粒子通过磁场时，力作用在粒子上，这一束粒子偏离了它的直线轨道。电流通过导体就像一束移动的电荷，因此能够被磁场偏移。这个现象是 E. R. Hall 在 1879 年发现的，因此称为**霍尔效应**（Hall effect）。考虑在导电盘上移动的电子受垂直磁场的作用（见图 2-23）。由于磁场的作用，移动电子偏转到盘的一侧，因此这边带负电，另一边相应地带正电。这样就在材料里形成了电场。带电粒子继续分离直到电场上作用的力与磁场上作用的力平衡。结果是横向的感应电动势 V 为

$$V = K_{H}\frac{BI}{t}$$

其中，B 是垂直射入平板的磁通量，I 是通过的电流，t 是平面的厚度，K_{H} 是常量叫作**霍尔系数**。因此如果有个恒流源使用了霍尔传感器，那么霍尔电压可以测量磁通量。

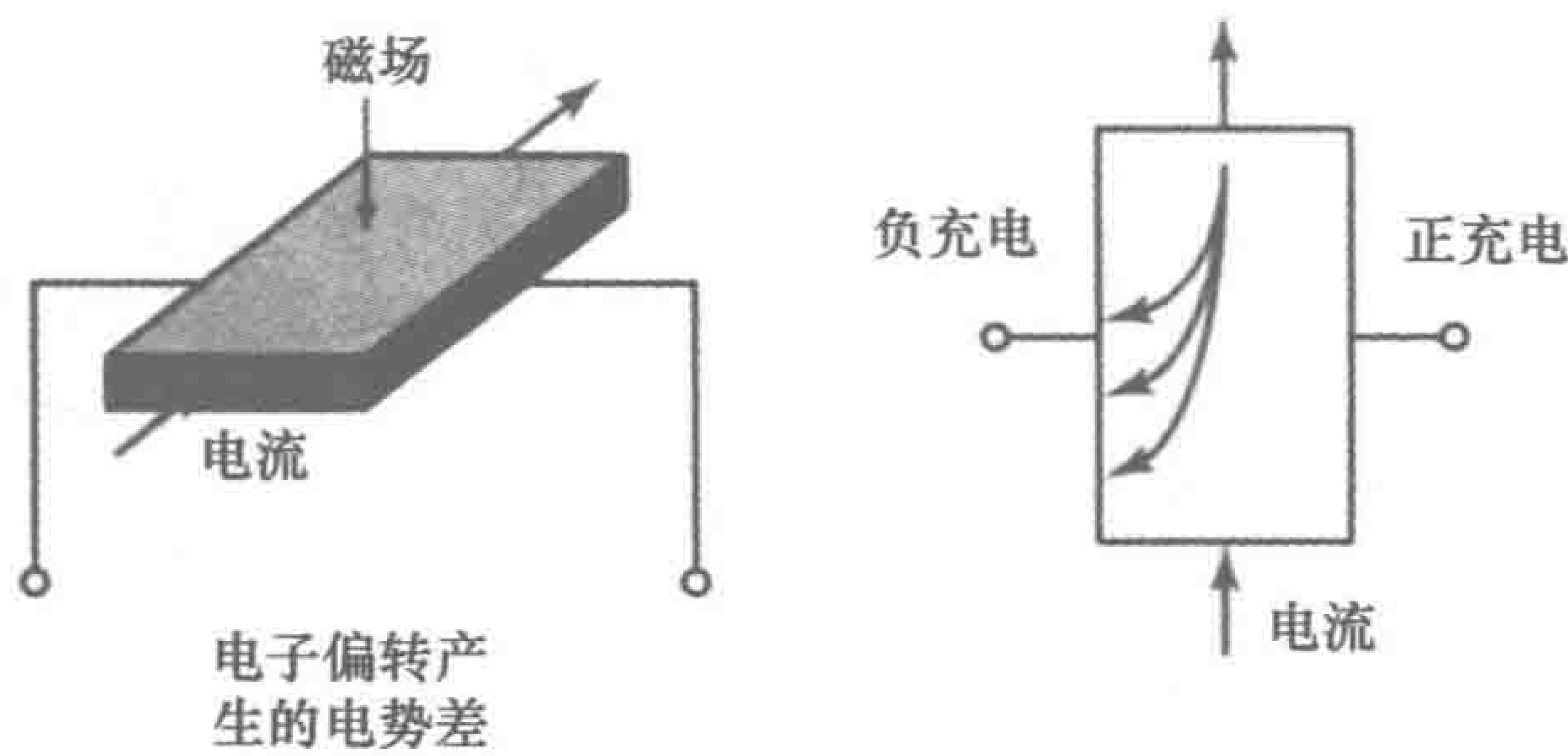


图 2-23 霍尔效应

霍尔效应传感器通常要与信号处理电路集成在一起。有两种基本形式的传感器：一种是线性的，输出和磁通量呈线性变化（见图 2-24a）；还有一种是开关式的，在某个特殊磁通量（见图 2-24b）下输出急剧下降。线性输出霍尔效应传感器 634SS2 在 −40 ~ +40mT 的范围内给出完全线性的输出，大约是 10mV/mT（1mV/gauss），电源电压是 5V。当磁通量大约在 3mT（30 高斯）时开关式霍尔效应传感器 Allegro UGN3132U 的输出从 0 变到大约 145mV。霍尔效应传感器的优势是作为开关能在 100kHz 的重复速率下工作，花费比电动机械式开关小，同时也不会有接触抖动和由此产生的一系列接触而非单个接触的问题发生。霍尔效应传感器不怕环境污染，因此能够用于极端恶劣的环境。

45

如果被测物体安装了一小块永磁铁，那么这样的传感器能够用于位置、位移和接近检测。作为实例，这样的传感器能够用于测量汽车油缸中的液位。磁铁放在浮块上，随着油的液位变化，浮块到霍尔传感器的距离也发生变化（见图 2-25）。霍尔电压的输出结果是浮块到传感器

的距离的度量，因此该传感器也能被用于测量液缸的液位。

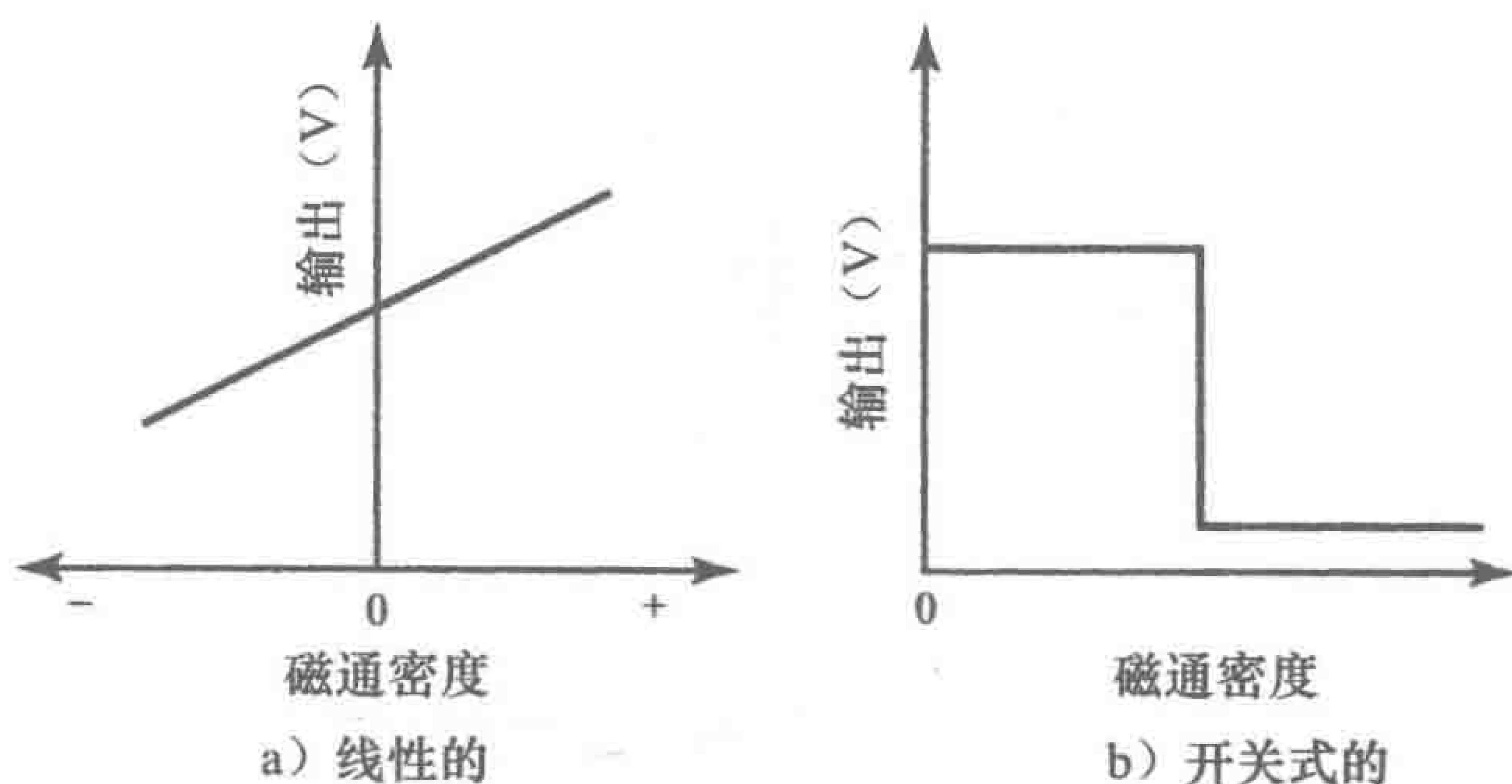


图 2-24 霍尔效应传感器

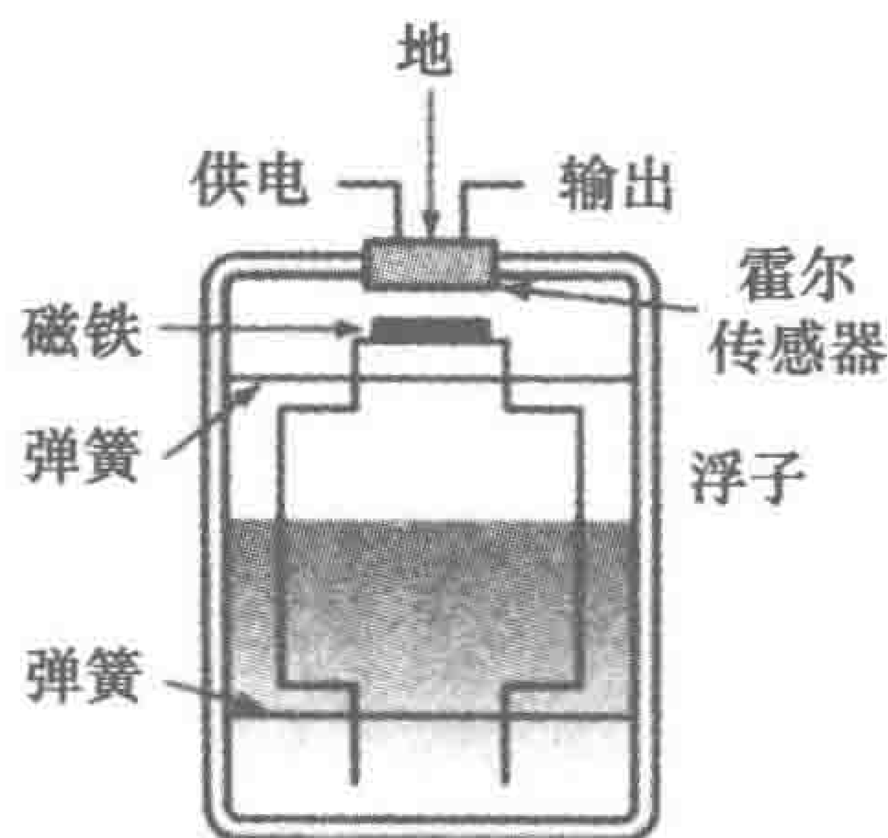


图 2-25 液位检测器

霍尔效应传感器的另一种应用是在无刷直流电动机方面。这样的电动机需要判定什么时候永磁铁转子正确地和定子上的绕组对齐，以便使电流在正确的时刻闭合通过绕组保持转子旋转。霍尔效应传感器用于检测何时它们是对齐的。

2.4 速度和运动

下面是用于监测线速度或者角速度并检测运动的传感器例子。运动检测器的应用还包括检测闯入者的安全系统、交互玩具和器械，比如，收银机屏幕在你接近它时会被激活。

2.4.1 增量式编码器

在 2.3.7 节描述过的增量式编码器能够用于检测角速度，这是由每秒产生的脉冲数决定的。

46

2.4.2 测速发电机

测速发电机用于测量角速度。一种形式是**可变磁阻测速发电机**，它由一个连在旋转轴上的磁性齿轮组成（见图 2-26）。耦合线圈绕在永久磁铁上。当轮子旋转时，轮子上的齿经过线圈，线圈和铁磁材料之间的气隙发生改变。我们会得到一个随着气隙周期性变化的磁路，因此通过耦合线圈的磁通量也发生变化。磁通量周期性变化的结果是在线圈中产生了一个交变的电动势（e. m. f.）。

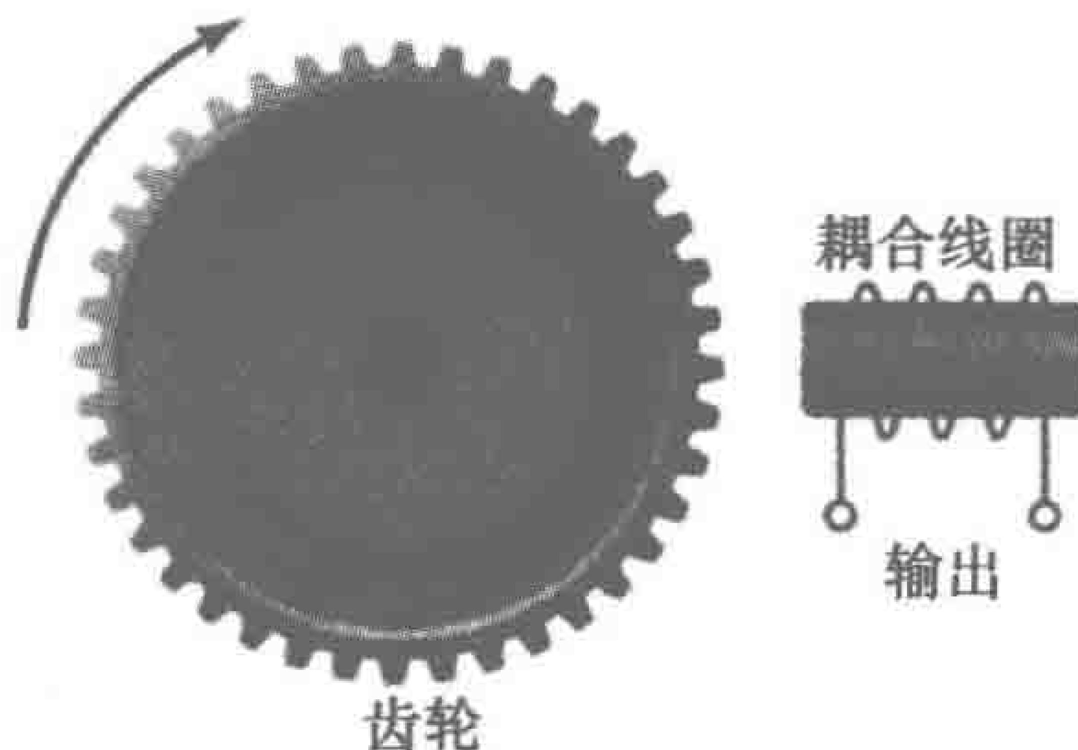


图 2-26 可变磁阻测速发电机

如果轮子包含 n 个齿，而且以角速度 ω 旋转，那么线圈的磁通量随着时间变化如下

$$\Phi = \Phi_0 + \Phi_a \cos n\omega t$$

其中，Φ_0 是磁通量的平均值，Φ_a 是磁通量变化的幅值。在 N 匝耦合线圈中的感应电动势 e 为 $-N\mathrm{d}\Phi/\mathrm{d}t$，因此

$$e = N\Phi_a n\omega \sin n\omega t$$

我们就可以写成

$$e = E_{max} \sin n\omega t$$

E_{max} 是感应电动势的最大值，大小为 $N\Phi_a n\omega$，所以是角速度的度量。

除了使用感应电动势的最大值测量角速度外，一个脉冲信号调理器能够把输出信号转换成能够被计数器计数的脉冲，在某个时间间隔内的脉冲数目能够用于测量角速度。

另一种形式的测速发电机本质上是**交流发电机**（a. c. generator）。它由一个称为转子的线圈

组成，能够绕着旋转轴旋转。线圈在稳定的永磁铁或者电磁铁产生的磁场中旋转（见图 2-27），感应出变化的电动势。感应电动势的幅值或频率可用于测量转子的角速度。输出可以整流为大小与角速度成正比的直流电压。这种传感器的非线性误差大约是满量程的 ±0.15% 数量级，通常使用于最高约 10 000rev/min 的转速测量。



图 2-27 交流发电机形式的测速发电机

47

2.4.3 热释电传感器

热释电（pyroelectric）材料，比如说钽酸锂（lithium tantalate），是响应热流产生电荷的结晶材料。当这种材料在电场中被加热到刚刚在居里（Curie）温度之下时，对钽酸锂来说是 610℃，然后在电场中冷却，材料中的电偶极子将排成线也就是被极化了（见图 2-8a、b）。当电场移去后，材料仍保持其极性。这个效应很像在磁场中将一块铁进行磁化。当热释电材料暴露在红外辐射中时，温度升高弱化了材料的极化量，电偶极子失去了原有排列的整齐性（见图 2-28c）。

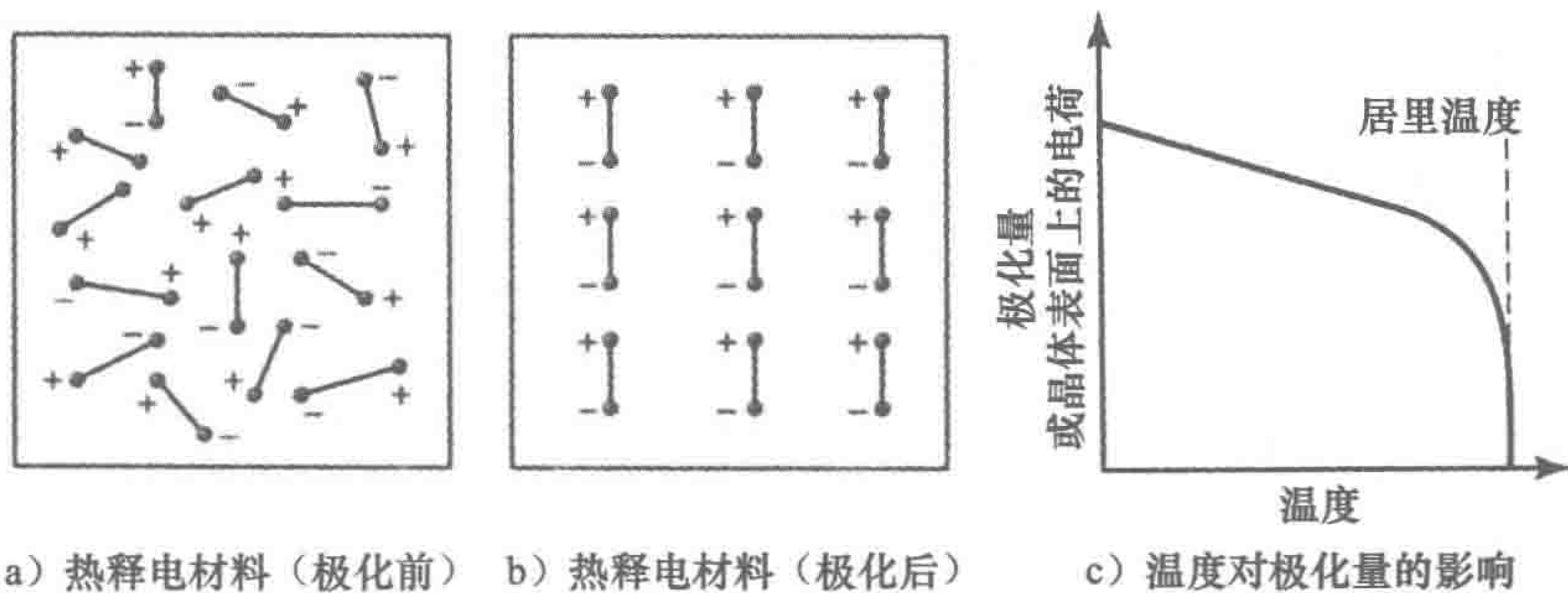


图 2-28 热释电材料被极化前后对比以及温度对极化量的影响

热释电传感器由被极化的热释电晶体和在正反面的金属薄膜电极组成。因为晶体带电表面被极化，所以从附近空气中分离出来的离子和任何连接到传感器上的测量回路中的电子都去平衡表面电荷（见图 2-29a）。如果有红外线辐射到晶体，并改变了它的温度，晶体的极化率将减小，相应地晶体表面的电荷也会减小。这样金属电极上的电荷会多于用来平衡晶体上的电荷所需要的电荷量（见图 2-29b）。电荷从测量回路泄漏直到晶体上的电荷又一次达到平衡。因此热释电传感器像一个电荷产生器一样，当红外线辐射导致温度变化时就会产生电荷。对图 2-28c的线性部分，当有一个温度变化，电荷变化 Δq 正比于温度变化 Δt：

48

$$\Delta q = k_p \Delta t$$

其中，k_p 是晶体的灵敏度常量。图 2-30 表示热释电传感器的等效回路，它实际上是一个被额外电荷充电的电容。电阻 R 要么表示内部的泄漏电阻，要么考虑外部电路的输入电阻。



图 2-29 热释电传感器

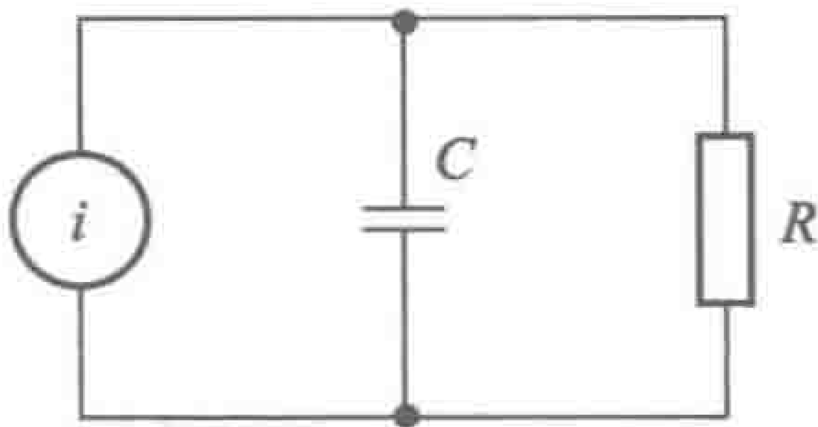


图 2-30 热释电传感器的等效回路

为了检测人类或者其他热源的运动，传感设备要区别背景的热辐射和移动热源的热辐射。单个的热释电传感器不能做到这一点，因此需要用两个（见图2-31）。一种形式是一个传感元件有一个单个前电极和两个分开的后电极。这样两个传感器能够相连，从而使得当两个传感器接收到相同的热信号时它们的输出就能抵消。当一个热源移动，这样热辐射就能从一个传感元件移动到另一个，最后导致的电流方向是通过电阻从第一个传感元件到第二个传感元件，运动方向相反电流方向就相反。一般一个移动的人给出的变化电流的数量级是 10^{-12}A。电阻 R 必须非常大才能给出一个明显的电压变化。例如，50GΩ 的电流产生的电压为 50mV。因此在电路中引入晶体管作为电压跟随器，使输出电阻降到几千欧。

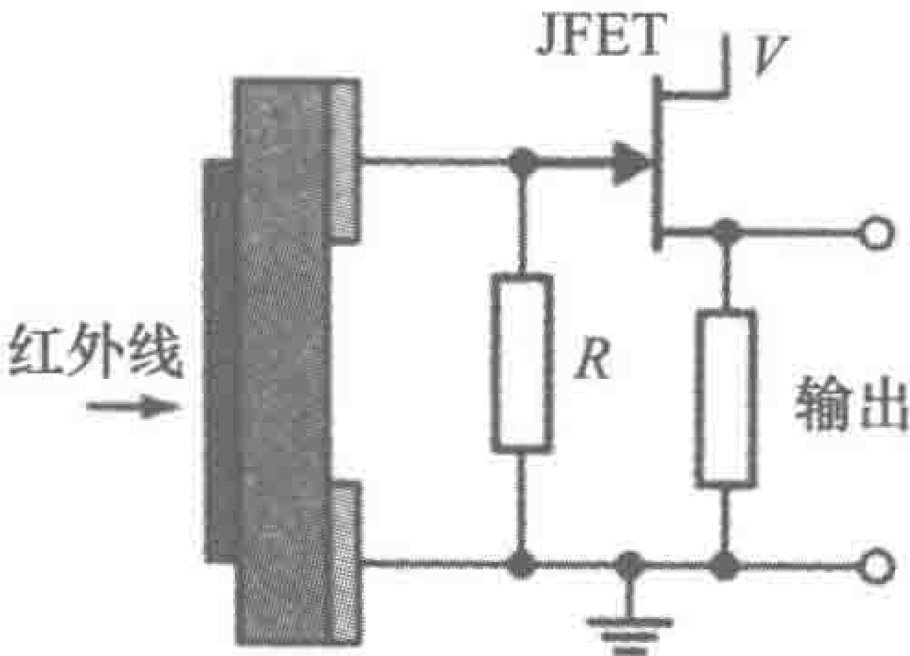


图2-31　双热释电传感器

还需要一个光学装置来聚焦传感器上的红外辐射信号。可以使用抛物面镜子，一个更常用的方法是使用菲涅尔（Fresnel）塑料透镜。这样的透镜能够保护传感器的前表面，同时也是触发闯入者警报或者当有人接近时把灯打开的传感器的常用形式。

2.5　力

弹簧秤是一个力传感器的例子，当力或者重物加载到秤盘上时引起位移，即弹簧伸展。这个位移可用于度量力的大小。力的测量通常用位移的测量来完成，下面的方法说明了这点。

2.5.1　应变式称重传感器

一种常用的测力传感器的形式是基于使用电阻应变片去监测在某个元件受力时产生拉伸、压缩和弯曲时的应变。这种装置通常称为**称重传感器**。图2-32展示了这样一个传感器的例子。这是一个贴有应变片的圆柱管，当有力施加在上面使其压缩时应变片的电阻根据应变大小产生变化，从而可以测量力的大小。由于温度同样会使电阻变化，所以需要使用信号调节电路来消除温度的影响（见3.5.1节）。通常这种称重传感器最大能测量到10MN，非线性误差大约是满量程的 ±0.03%，可重复误差和迟滞误差分别是满量程的 ±0.02%。基于弯曲应变的应变式称重传感器通常用于测量比较小的力，比如说变化范围是从0～5N最高到0～50kN。误差通常为非线性误差，是满量程的 ±0.03%，迟滞误差为满量程的 ±0.02%，可重复误差为满量程的 ±0.02%。

49



图2-32　应变式称重传感器

2.6　流体压力

很多在工业过程中检测流体压力的装置是需要检测膜片、膜盒、波纹管和波登管的弹性变形。它可以测量以下类型的压力：绝对压力（absolute pressure），测量的是相对于压力为0时的压力，即相对于真空的压力；差压（differential pressure），测量的是两个压力之差；表压（gauge pressure），测量的是相对于大气压的压力。

对于图2-33a、b所示的膜片，当两边有不同的压力时膜片的中心便产生了位移。膜片上的皱纹会产生更大的敏感性。这种运动能够被某种位移传感器检测到，比如图2-34中的应变片。通常使用特殊设计的应变片。考虑有4个应变片设计的情况，其中两个测量周向的应变，

两个测量径向的应变。4 个应变片连成惠斯通电桥（Wheatstone bridge）臂的形式（见第 3 章）。相对于应变片粘在膜片上，一个替代的方法是建立以特定掺杂区域为应变片的硅膜片。这样的安排通常用于汽车的电子系统检测进气管道的压力。

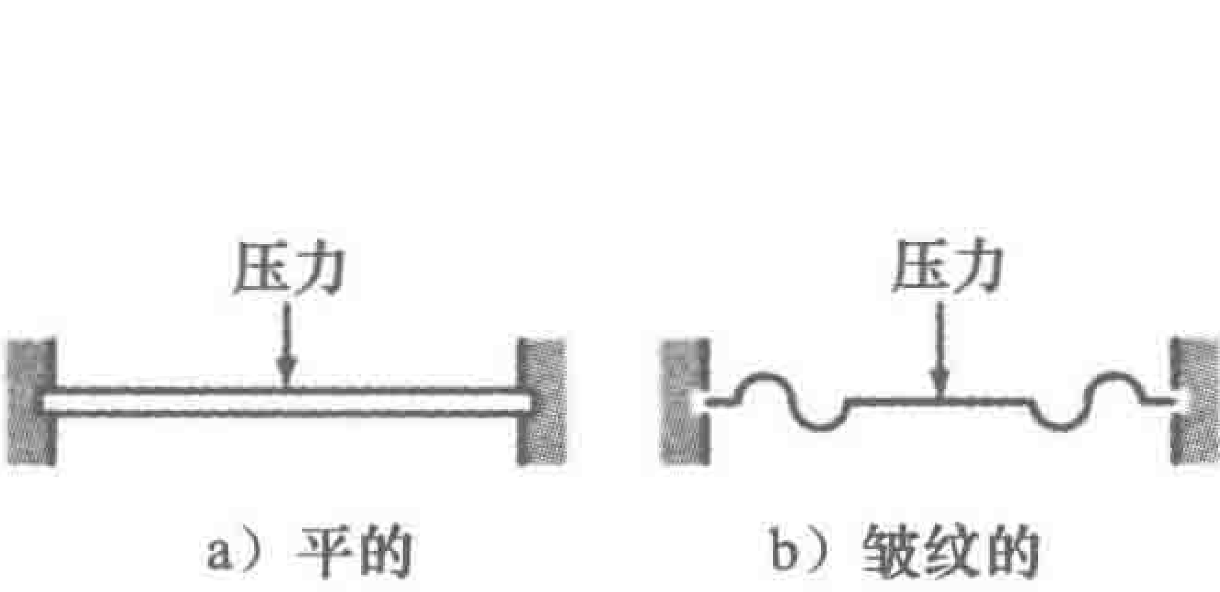


图 2-33　膜片

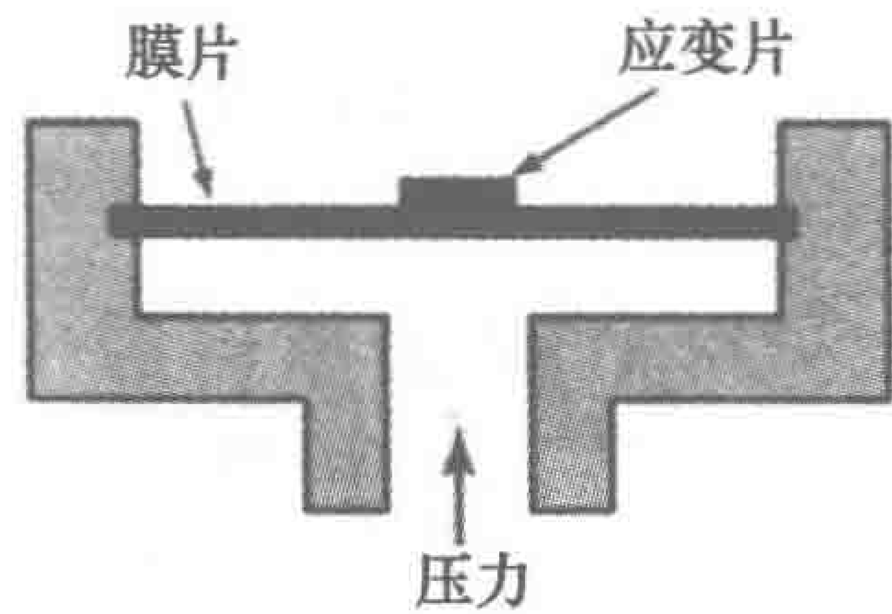


图 2-34　膜片压力应变片

对于摩托罗拉的 MPX 压力传感器，应变片元件和一个电阻网络集成在一个单硅膜片芯片中。当电流通过应变片元件且压力垂直施加在上面时，就会产生电压。应变片元件、信号调理和温度补偿电路一起封装为 MPX 传感器。输出电压和压力成正比。这样的传感器可用于测量绝对压力（以 A、AP、AS 或者 ASX 为尾缀的 MX 编号系统）、压差（以 D 或者 DP 为尾缀的 MX 编号系统）和表压（以 GP、GVP、GS、GVS、GSV 或者 GVSX 为尾缀的 MX 编号系统）。例如，MPX2100 系列的压力范围是 100kPa，直流电源电压为 16V，给出在 40mV 范围内的绝对压力和压力差。对于 0 ~ 100kPa 的阶跃变化，响应从 10% 上升到 90%，响应时间为 1.0ms，输出电阻的数量级为 1.4 ~ 3kΩ。绝对压力传感器用于高度表和气压计，压差传感器用于气流测量，表压传感器用于发动机压力和胎压的测量。

50

膜盒（见图 2-35a）可以认为是两个有波纹的膜片连在一起，因此有更大的灵敏度。多个膜盒连在一起就是波纹管，如图 2-35b 所示，有更高的灵敏度。图 2-36 显示的是波纹管如何与一个线性差动变压器结合在一起，从而给压力传感器一个电量输出。膜片、膜盒和波纹管是用不锈钢、磷青铜和镍制成的，橡胶和尼龙也用于某些膜片。在 10^3 ~ 10^8Pa 范围内的压力能够被这样的传感器检测到。

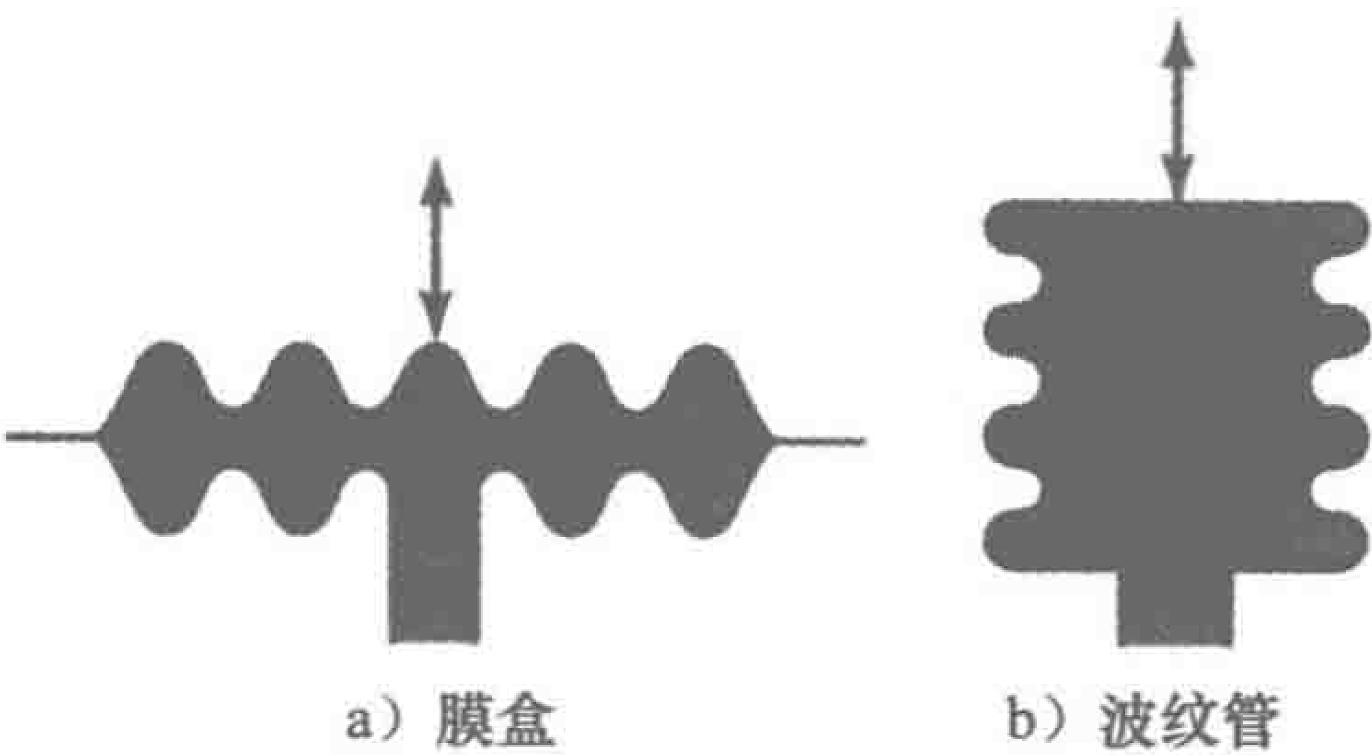


图 2-35　膜盒和波纹管

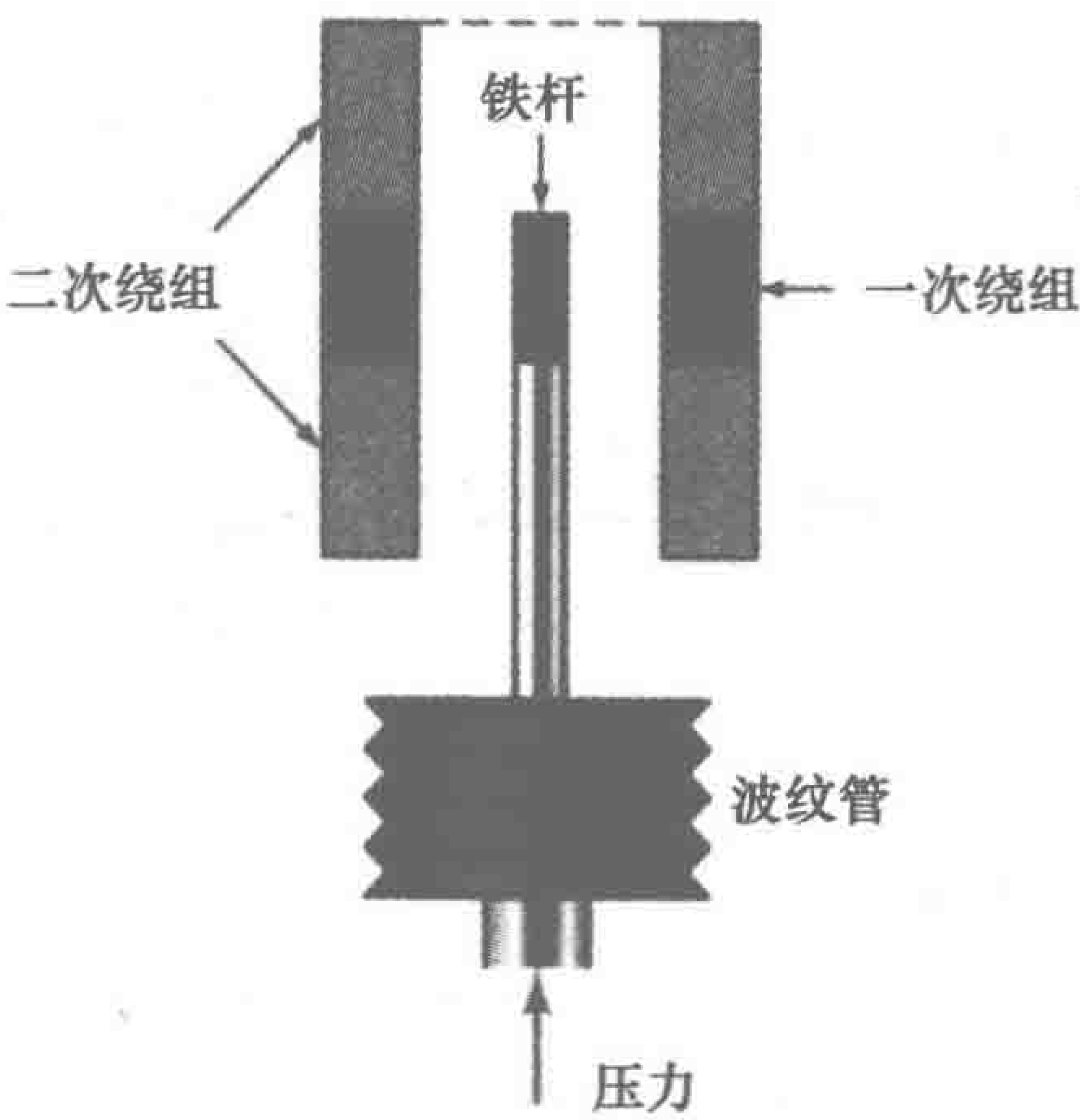


图 2-36　有波纹管的 LVDT

用椭圆截面的管子可以得到不同形式的变形（见图 2-37a）。随着压力的增长使这样的管子趋于圆截面。当这样的管子是如图 2-37b 所示的 C 型时，一般称为**波登管**（Bourdon tube），当压力增长 C 的开口就增大。螺旋形的管子（见图 2-37c）有更大的灵敏度。管子由不锈钢和

51 磷青铜制成，用于测量 $10^3 \sim 10^8$Pa 的压力。

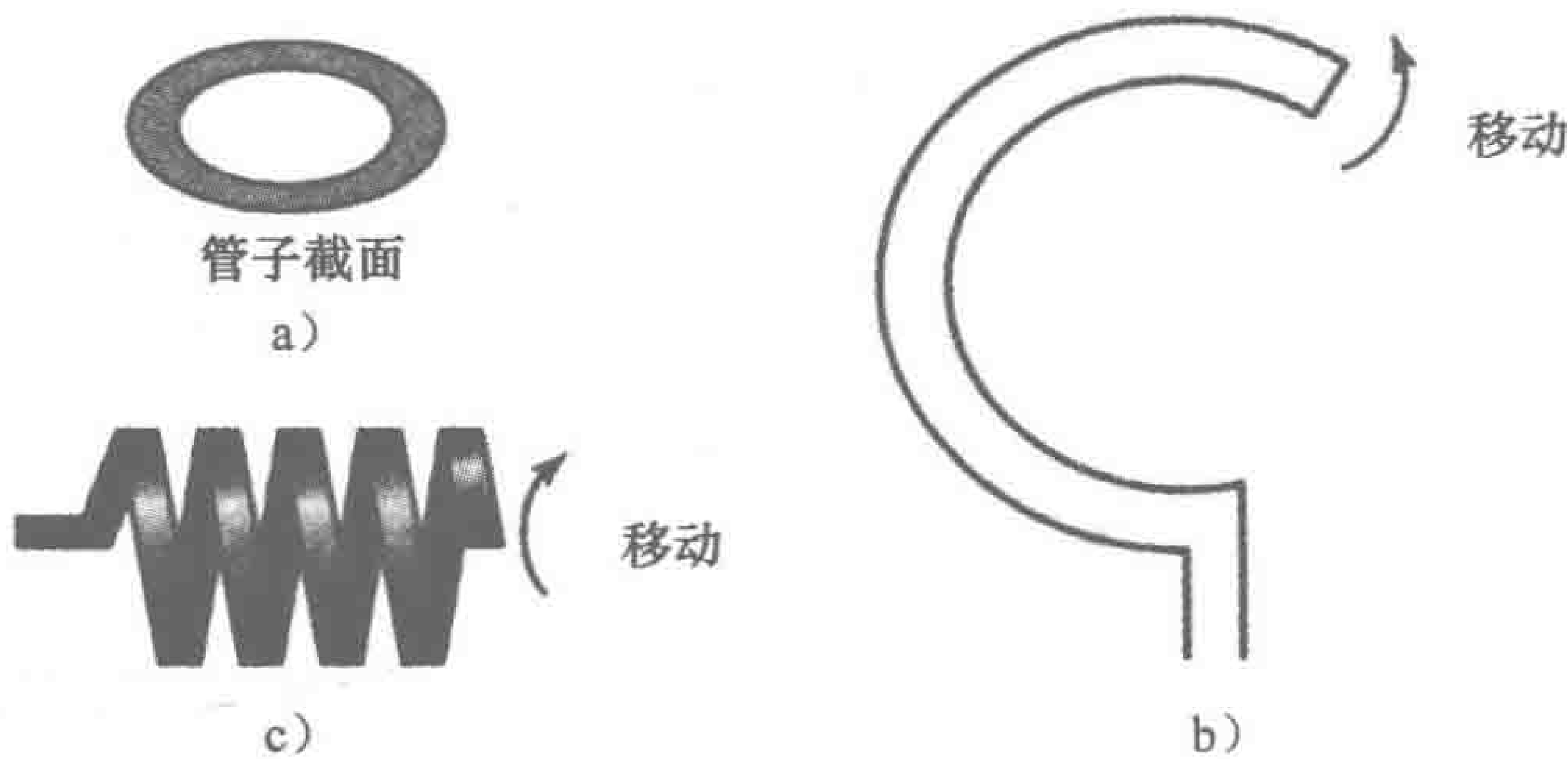


图 2-37 管状压力传感器

2.6.1 压电传感器

压电材料是当被拉伸和挤压时会使材料的一面带正电，另一面带负电（见图 2-38a）。因此就产生了电压。压电材料是离子晶体，当拉、压时会导致在晶体的电荷分布发生变化，使电荷发生移动。表面上的净电荷 q 正比于电荷移动的量 x，由于该移动量正比于施加的力 F，因此有

$$q = kx = SF$$

其中，k 是常量，S 也是称为**电荷灵敏度**（charge sensitivity）的常量。电荷灵敏度取决于材料和晶体的方向。当石英（quartz）是在某个特定的方向被切断且施加的力也在某个特定的方向时，石英的电荷灵敏度为 2.2pC/N。钛酸钡（barium titanate）有更高的电荷灵敏度，数量级为 130pC/N，锆钛酸铅（lead zirconate-titanate）的电荷灵敏度大约是 265pC/N。

金属电极放在压电材料的两面（见图 2-38b）。在两板之间压电材料的电容 C 是

52

$$C = \frac{\varepsilon_0 \varepsilon_r A}{t}$$

ε_r 是材料的相对介电常量，A 是面积，t 是厚度。由于电荷 $q = Cv$，v 是电容两边的电势差，所以

$$v = \frac{St}{\varepsilon_0 \varepsilon_r A} F$$

F 是施加在面积 A 上的力，所以压力 p 为 F/A。如果我们写 $S_v = (S/\varepsilon_0 \varepsilon_r)$，即**电压敏感系数**，则有

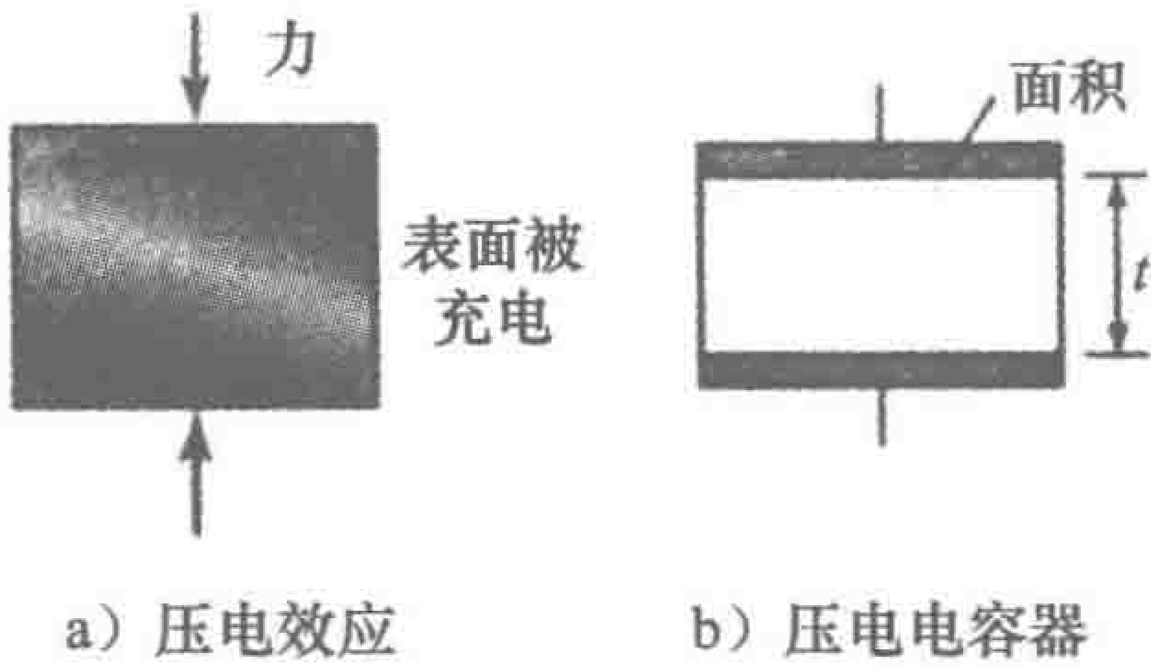


图 2-38 压电效应和压电电容器

$$v = S_v tp$$

电压与施加的压力成正比。石英的电压敏感系数大约为 0.055V/m Pa。钛酸钡大约是 0.011V/m Pa。

压电传感器用于测量压力、力和加速度。由于压力产生的电荷很快就泄漏完，所以应用大多是瞬态的而不是稳定的压力。压电传感器的等效电路是电荷产生器与电容 C_s 和电阻 R_s 的并联，R_s 是由于电荷通过电介质泄漏时产生的电阻（见图 2-39a）。当传感器通过一个电容 C_c 的电缆连接到输入电容是 C_A、电阻是 R_A 的放大器时，我们就得到了如图 2-39b 所示的等效电路，整个电路的电容为 $C_s + C_A + C_c$，电阻是 $R_A R_s/(R_s + R_A)$，且两者并联。当传感器受到压力后会带电，但由于电阻将使电容随着时间放电。放电的时间取决于电路的时间常量。

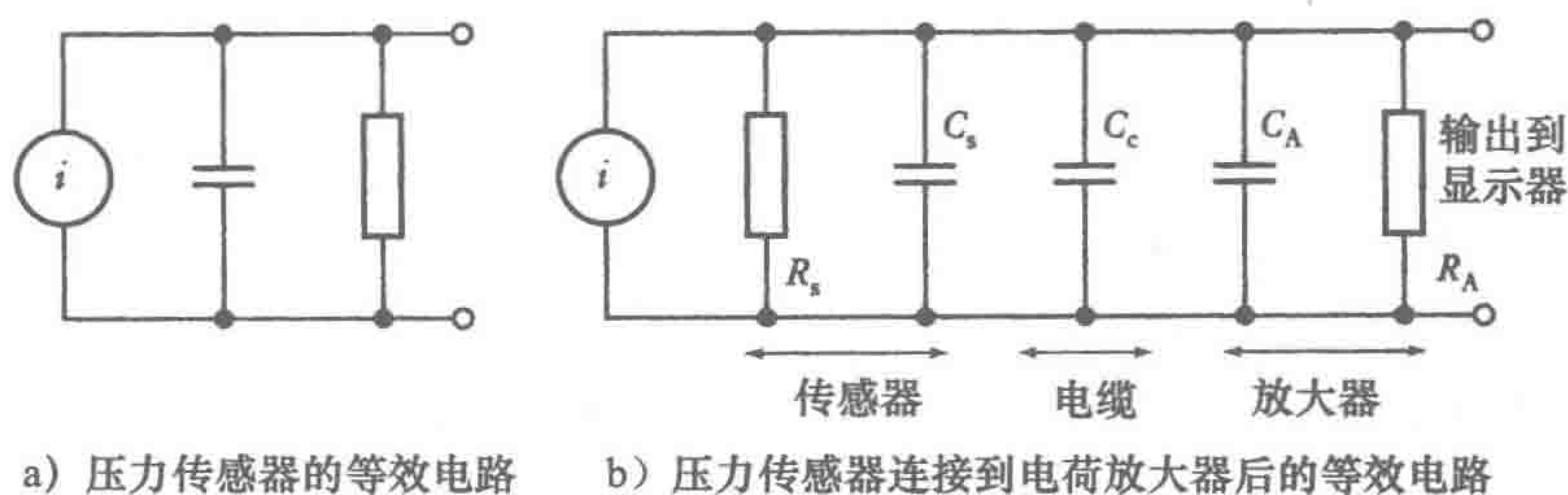


图2-39 压力传感器的等效电路

2.6.2 触觉传感器

触觉传感器是压力传感器的特殊形式。这样的传感器用于判断机器人的手的指尖是否接触到某个物体。它们也可以用于感受身体接触的触摸屏。一种形式的触觉传感器使用了压电聚偏二氟乙烯(Piezoelectric Polyvinylidene Fluoride，PVDF）膜片。它使用中间隔着能传播振动的软膜的两层膜片（见图2-40)。下面的PVDF通上交变的电压，会导致膜片的机械振动（逆压电效应)。中间的膜片把振动传到上面的PVDF。由于压电效应，这些振动会通过上面的膜片产生变化的电压。当施加压力时，上面的PVDF膜片的振动受到影响，输出电压会被改变。

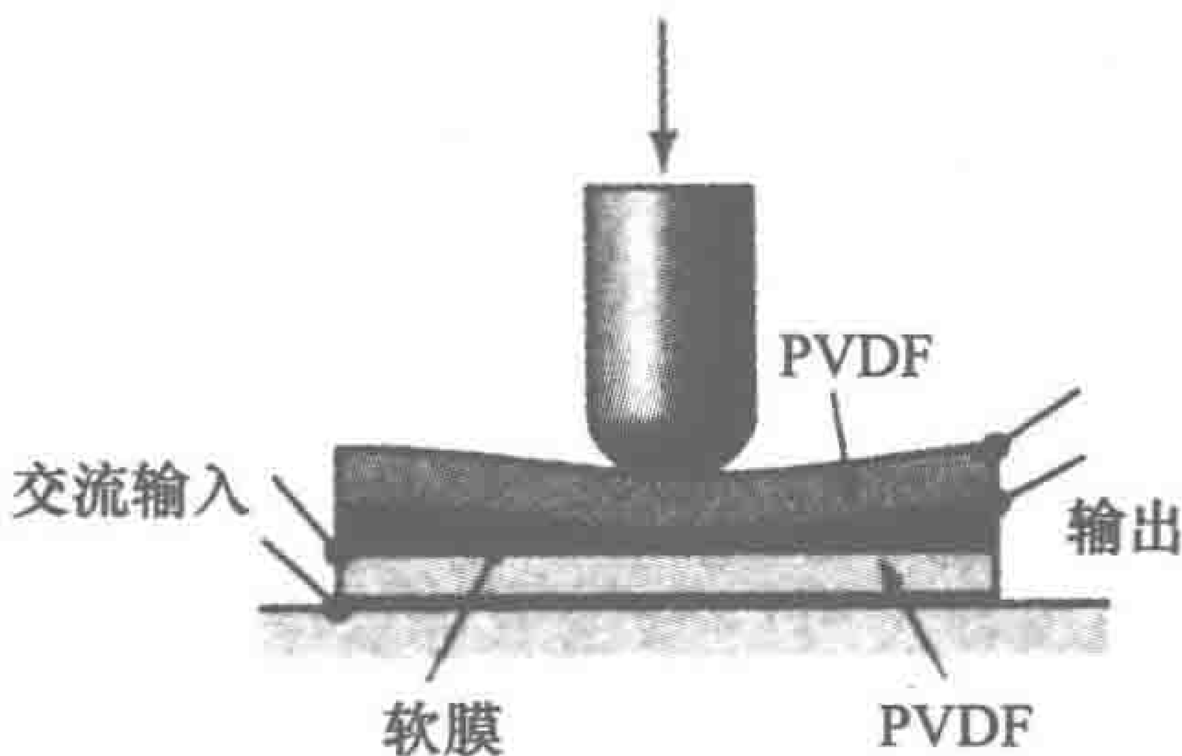


图2-40 PVDF触觉传感器

2.7 液体流动

传统的测量液体流动的方法是用一个装置来测量当液体流过一个颈缩管时候的压降（见图2-41)。对于水平管，v_1 是压缩前的流体速度，p_1 是压缩前的压力，A_1 是管子压缩前的横截面积，v_2 是管子压缩后的流体速度，p_2 是压缩后的压力，A_2 是压缩后的面积，液体密度是 ρ，则伯努利方程为：

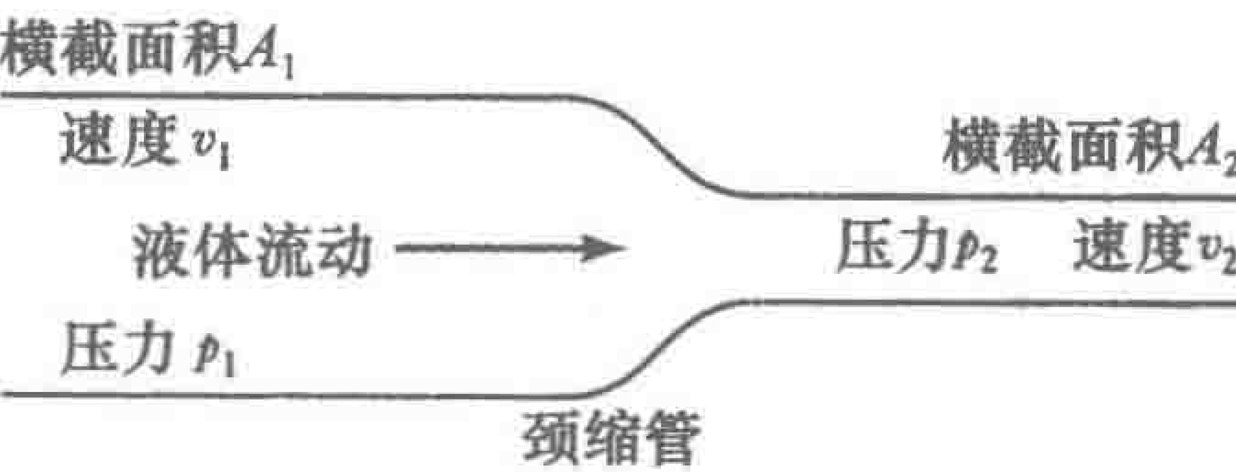


图2-41 液体流过一个颈缩管

$$\frac{v_1^2}{2g}+\frac{p_1}{\rho g}=\frac{v_2^2}{2g}+\frac{p_2}{\rho g}$$

由于每秒通过没压缩之前的管子的液体质量等于流过压缩后的管子的液体质量，所以得到 $A_1v_1\rho=A_2v_2\rho$。但是每秒通过管子的液体流量Q是 $A_1v_1=A_2v_2$。因此

$$Q=\frac{A_2}{\sqrt{1-(A_2/A_1)^2}}\sqrt{\frac{2(p_1-p_2)}{\rho}}$$

因此，每秒通过的液体流量 Q 正比于压力差的平方根。于是，测量压力差能够测出流体的速率。很多装置都是基于这个原理，下面的孔板可能就是最常用的例子之一。

2.7.1 孔板

孔板（见图2-42）就是很简单的一个盘，中间有一个孔，放在流动液体的管子里。压力差是从管子上游距离为管直径大小的一个点到管子下游的距离为

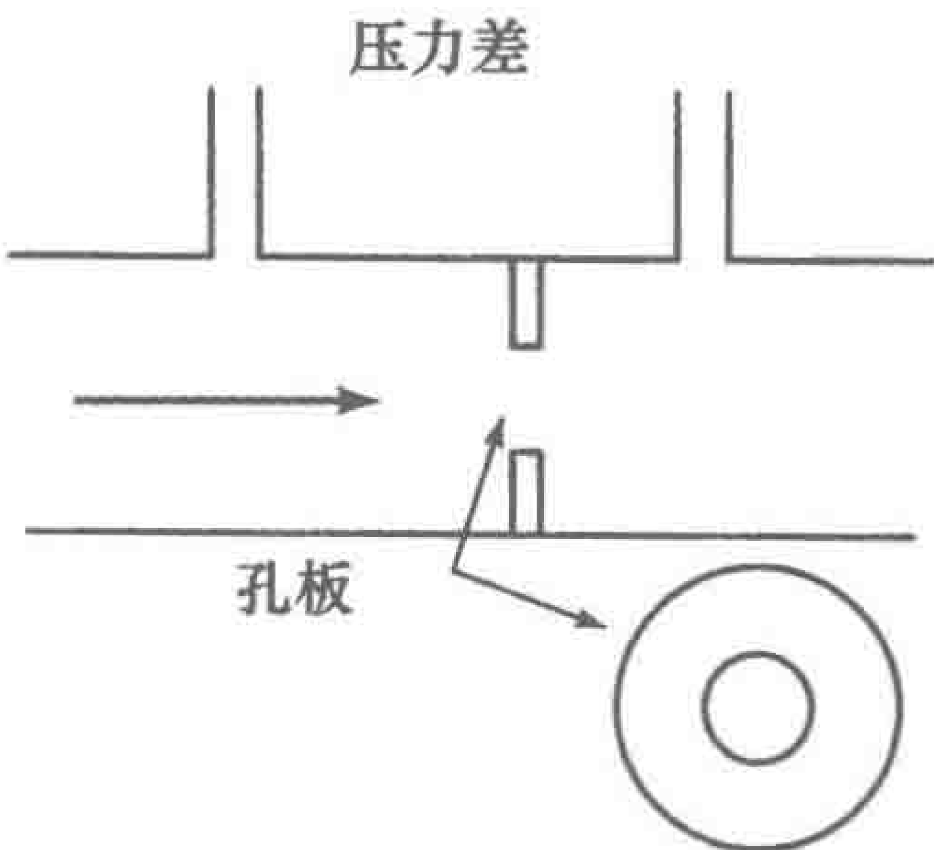


图2-42 孔板

53
54

管半径大小的一个点。孔板简单、便宜而且没有移动的部分，因而被广泛地使用。精度通常为满量程的 ±1.5%，它是非线性的，会在它所连接的系统中产生一个相当的压力损失。

2.7.2　涡轮流量计

涡轮流量计（见图 2-43）由支撑在液体流动中心的多叶的转子组成。液体流动会导致转子旋转，角速度和流速近似成正比。转子的转速可以由磁场采集器检测。记录脉冲数就可以确定转子的转速。这个流量计比较贵，但是精度一般为 ±0.3%。

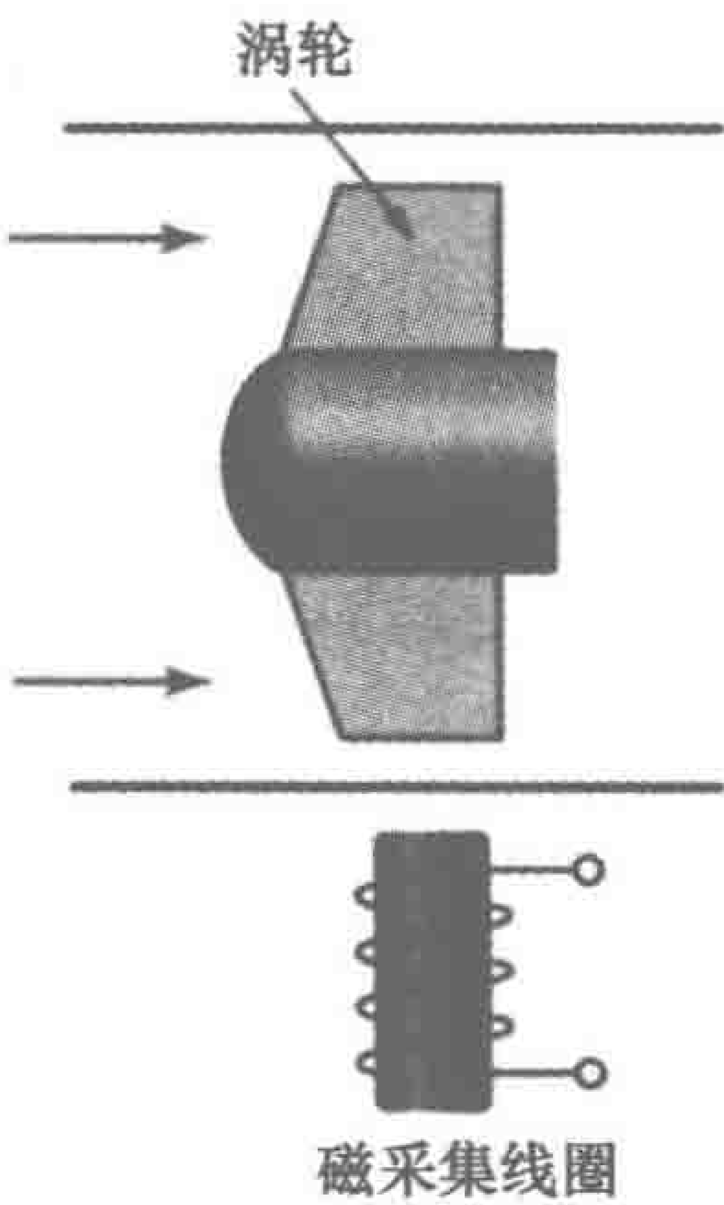


图 2-43　涡轮流量计

2.8　液位

容器中的液位可以通过直接检测液体表面的位置或者间接地检测其他与高度有关的变量来检测。直接检测的方法需要浮块；间接检测方法包括检测容器的重量，比如通过承重传感器来检测。液体的重量为 $Ah\rho g$，其中 A 是容器的横截面积，h 是液体的高度，ρ 是液体密度，g 是重力加速度。因此高度的变化导致重量的变化。更常用的间接方法可以测量液体中某个点的压力，由 h 高的液体产生的压力为 $h\rho g$，其中 ρ 是液体密度。

2.8.1　浮块

直接检测容器液位的方法就是检测浮块的移动。图 2-44表示了一个简单的浮块系统。浮块的位移导致杠杆臂旋转，然后使电位计的滑块移动，从而输出和液位相关的电压。其他形式的浮块测量系统则是推动杠杆带动线性可变差动变压器的磁心发生位移，或者拉、压应变片元件。

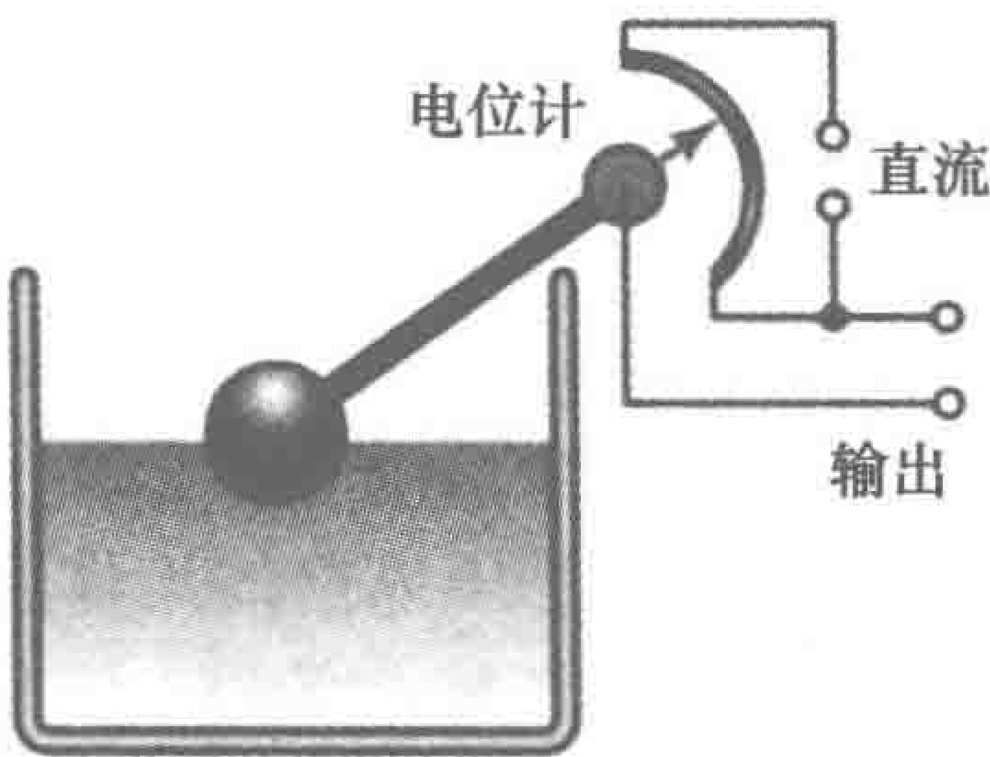


图 2-44　浮块系统

2.8.2　压差

图 2-45 显示的是两种通过测量压差来测量液位的形式。在图 2-45a 中，差压传感器确定了容器底座的液体压力和容器通向大气的开口的气压之间的压差。图 2-45b 的装置在开放和关闭容器中都能使用，压差装置测量的是容器底座压力和液体上表面的空气或者气体之间的差压。

55

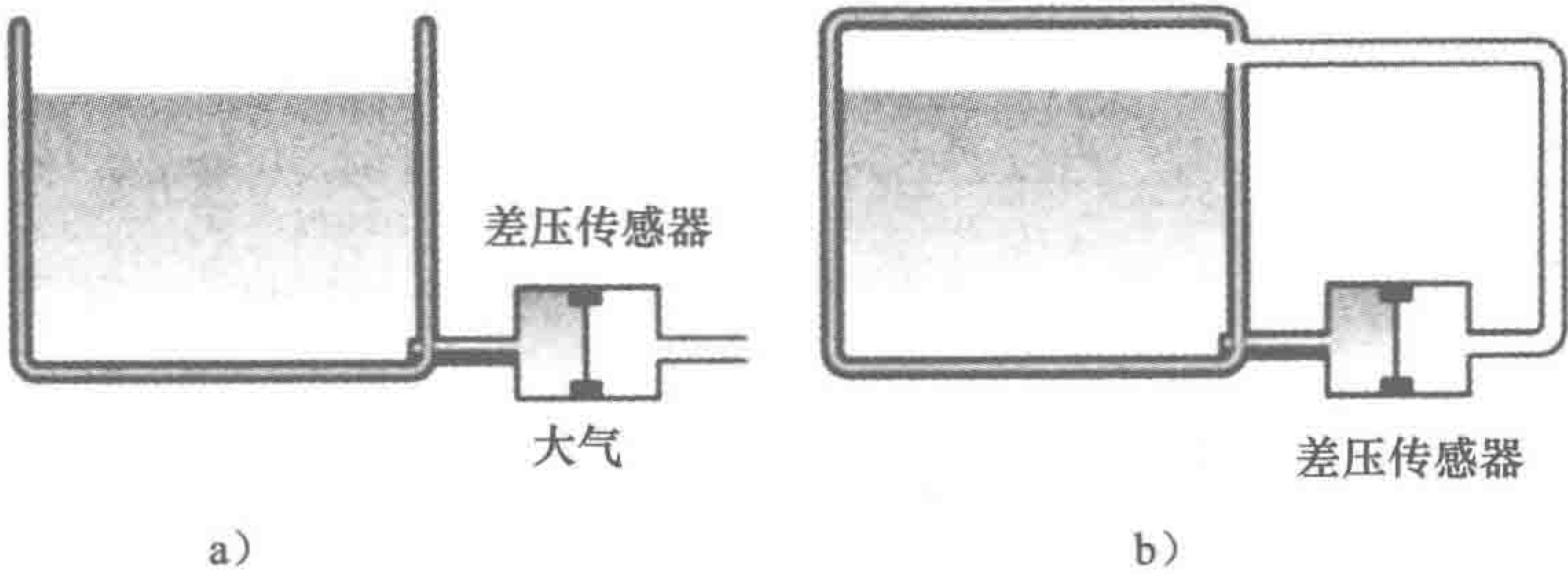


图 2-45　使用差压传感器

2.9　温度

检测温度通常使用的方法都是检测固体、液体或者气体的膨胀或收缩，导体或者半导体的

电阻变化和热释电材料的感应电动势变化。下面是一些温度控制系统的常用测量方法。

2.9.1 双金属片

这种装置是两个不同的金属条连接在一起组成的。不同金属有不同的膨胀系数，当温度发生变化时，复合在一起的金属条会形成一条曲线，系数高的金属在曲线的外面。这个变形可以用于温度控制开关，在家庭中常用的简单的恒温器中使用（见图 2-46）。图中的小磁块使传感器表现出迟滞，这意味着开关闭合和断开的温度不同。

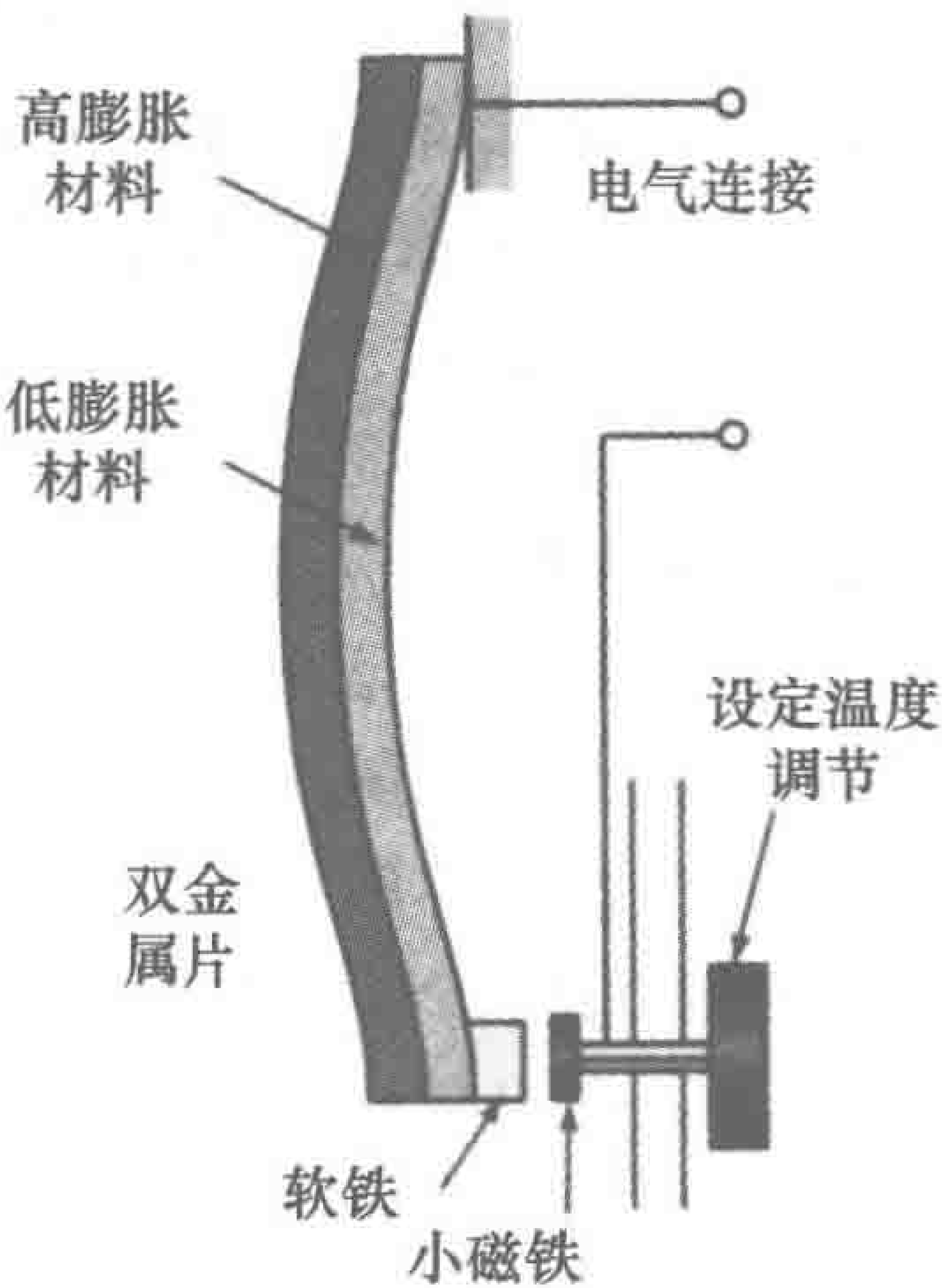


图 2-46 双金属片恒温器

2.9.2 电阻温度检测器

在有限的温度范围内，大多数金属电阻阻值的增加和温度是线性关系（见图 2-47）。该线性关系是：

$$R_t = R_0(1 + \alpha t)$$

其中 R_t 是在 t℃时的电阻，R_0 是在 0℃时的电阻，α 是金属的常量叫作电阻温度系数。电阻温度检测器（Resistance Temperature Detector，RTD）是简单的电阻元件，通常为铂、镍或者镍铜合金等金属导线的线圈形式，铂是最常用的。薄膜铂元件通常是在衬底镀上一层金属制成，铂线制成的绕线元件需要用高温的玻璃胶固定在陶瓷管内。这样的检测器性能很稳定，且能在很长时间间隔给出重复的响应。它们响应实现的数量级为 0.5 ~ 5s 或者更多。

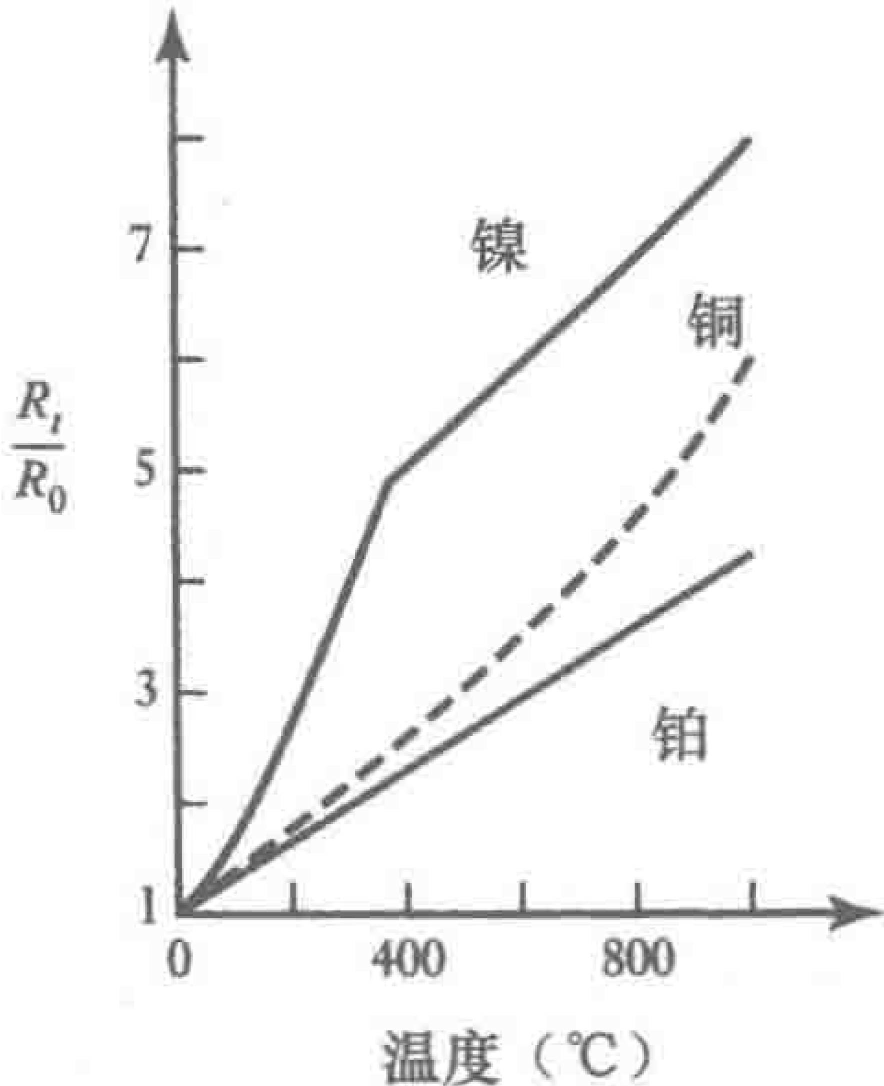


图 2-47 金属电阻随着温度的改变

56

2.9.3 热敏电阻

热敏电阻是由小片的金属氧化物的混合物材料做成的，比如铬、钴、铁、锰和镍。这些氧化物都是半导体。这些材料被做成不同形式的元件，比如珠、盘和棒（见图 2-48a）。

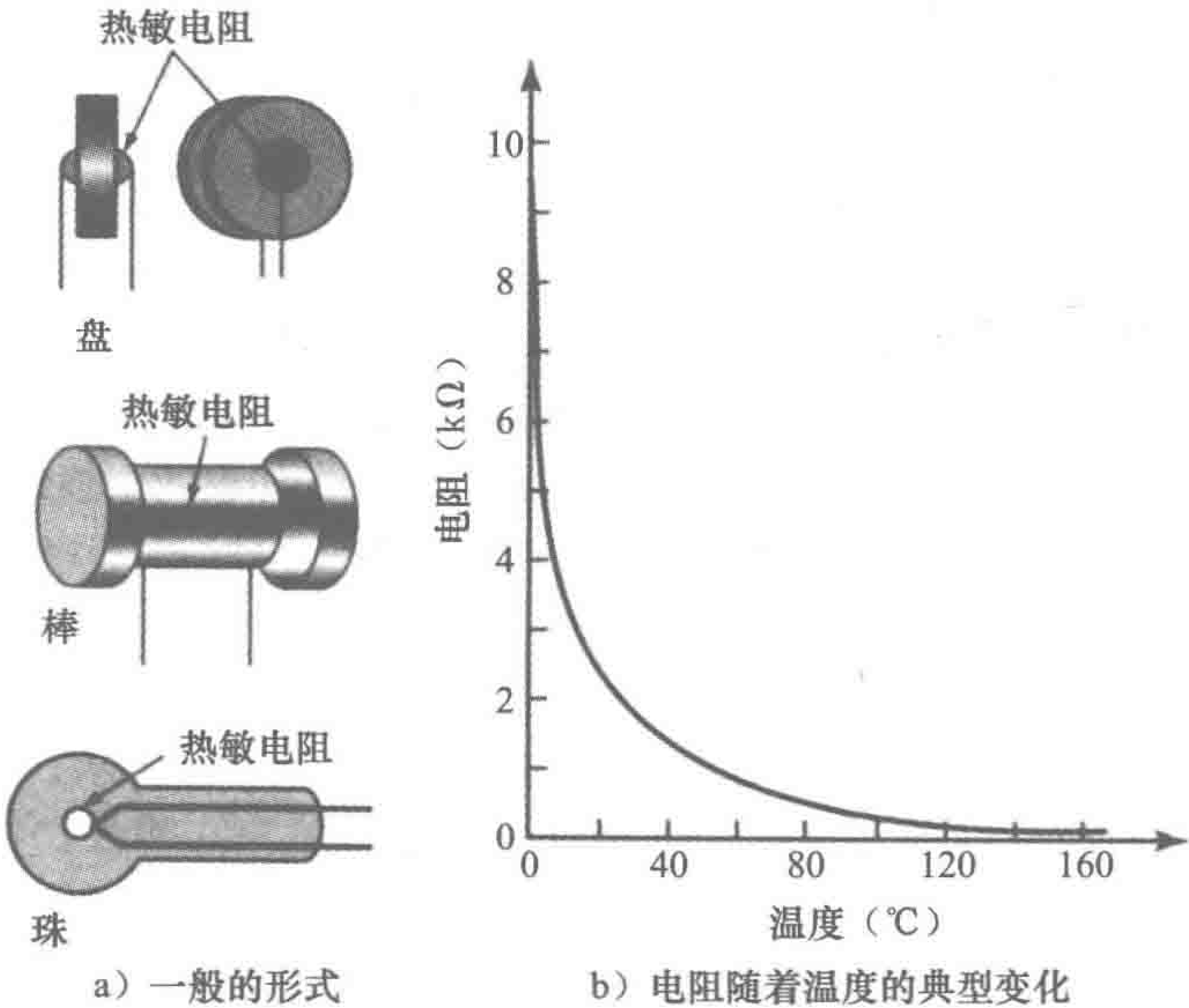


图 2-48 热敏电阻

传统的金属氧化物的热敏电阻的阻值随着温度的增加以很大的非线性减小（见图2-48b）。这样的热敏电阻有负的温度系数（Negative Temperature Coefficient，NTC）。然而，也有正的温度系数（Positive Temperature Coefficient，PTC）的热敏电阻。每度温度变化带来的热敏电阻阻值变化比金属的阻值变化大很多。热敏电阻的温度电阻关系可以用下面形式的公式来描述：

$$R_t = Ke^{\beta/t}$$

其中，R_t 是在 t℃时的电阻，K 和 β 是常量。热敏电阻相对于其他形式的温度传感器有很多优势。它们很坚固而且很小，这样保证传感器能够对某一有效点进行温度测量。由于它们体积很小，它们对温度变化的响应速度很快。温度变化一度会使它们的电阻产生很大变化。它们主要的缺点是非线性。热敏电阻在汽车电子系统中用于检测空气温度和冷却空气温度的变化。

57

2.9.4　热敏二极管和晶体管

一个p-n结半导体二极管被广泛地作为温度传感器使用。当掺杂的半导体温度变化时，它们的电荷载流子的移动能力也发生变化，这就影响了电子和空穴通过p-n结扩散的速率。因此当p-n结中间有电势差 V 时，通过p-n结的电流是一个关于温度的方程：

$$I = I_0(e^{eV/kT} - 1)$$

其中，T 表示热力学温度，e 是电子的电荷，k 和 I_0 是常量。对方程取对数我们就得到了电压 V 的方程：

$$V = \left(\frac{kT}{e}\right)\ln\left(\frac{I}{I_0} + 1\right)$$

因此，对于一个恒定电流，我们发现 V 正比于热力学温度，因此通常用恒定电流的二极管两极的电势差测量温度。这种传感器很像热敏电阻，但是它的优势是对温度的响应是线性的。用于温度传感器的二极管和必要的信号处理电路，通常以集成电路的形式一起提供，比如说LM3911，从而形成很紧凑的传感器。LM3911 的输出电压正比于变化速率为10mV/℃的温度。

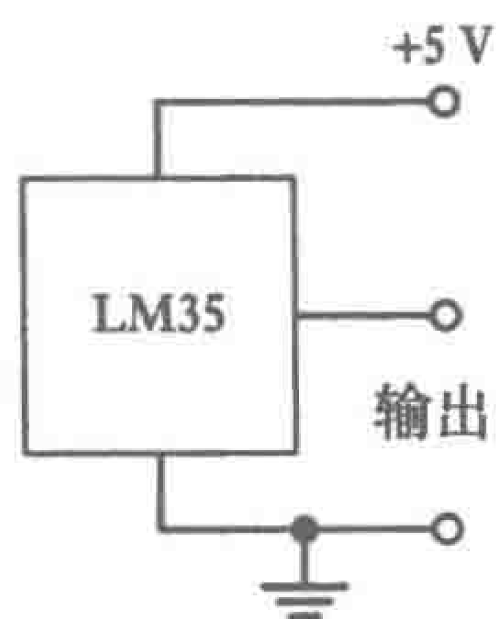


图2-49　LM35

与热敏二极管类似，热敏晶体管在基极和发射极之间的结电压取决于温度，它可以用于测量温度。一个常用的方法是测量两个不同集电极电流的晶体管的基极和发射极之间电压的差值，这个差值直接正比于热力学温度。这样的晶体管可以和其他电路元件集成在一块芯片上，从而形成带有信号调理功能的温度传感器，例如LM35（见图2-49）。这样的传感器用于 -40 ~ 110℃的测量范围，并给出10mV/℃的输出。

2.9.5　热电偶

如果两个不同的金属连在一起，在连接处会有电势差。电势差的大小取决于使用的金属和连接处的温度。一个热电偶是一个带有两个这样连接端点的完整电路（见图2-50a）。

58

如果两个连接端点有相同的温度，那么就没有净电动势。如果两个连接端点有温度差，就有电动势。电动势的值 E 取决于连接的两个金属和两个连接端的温度 t。通常一个连接端是0℃，在一个合理范围内，下面的关系式成立：

$$E = at + bt^2$$

其中，a 和 b 是所用金属有关的常量。通常使用的热电偶如表2-1所示，其中列出了常用的温度范围和常用的灵敏度。这些常用的热电偶以参考字母给出。例如，铁/康铜（iron/constantan）热电偶称为J型热电偶。图2-50b显示了常用的几对金属的电动势如何随着温度变化的曲线。

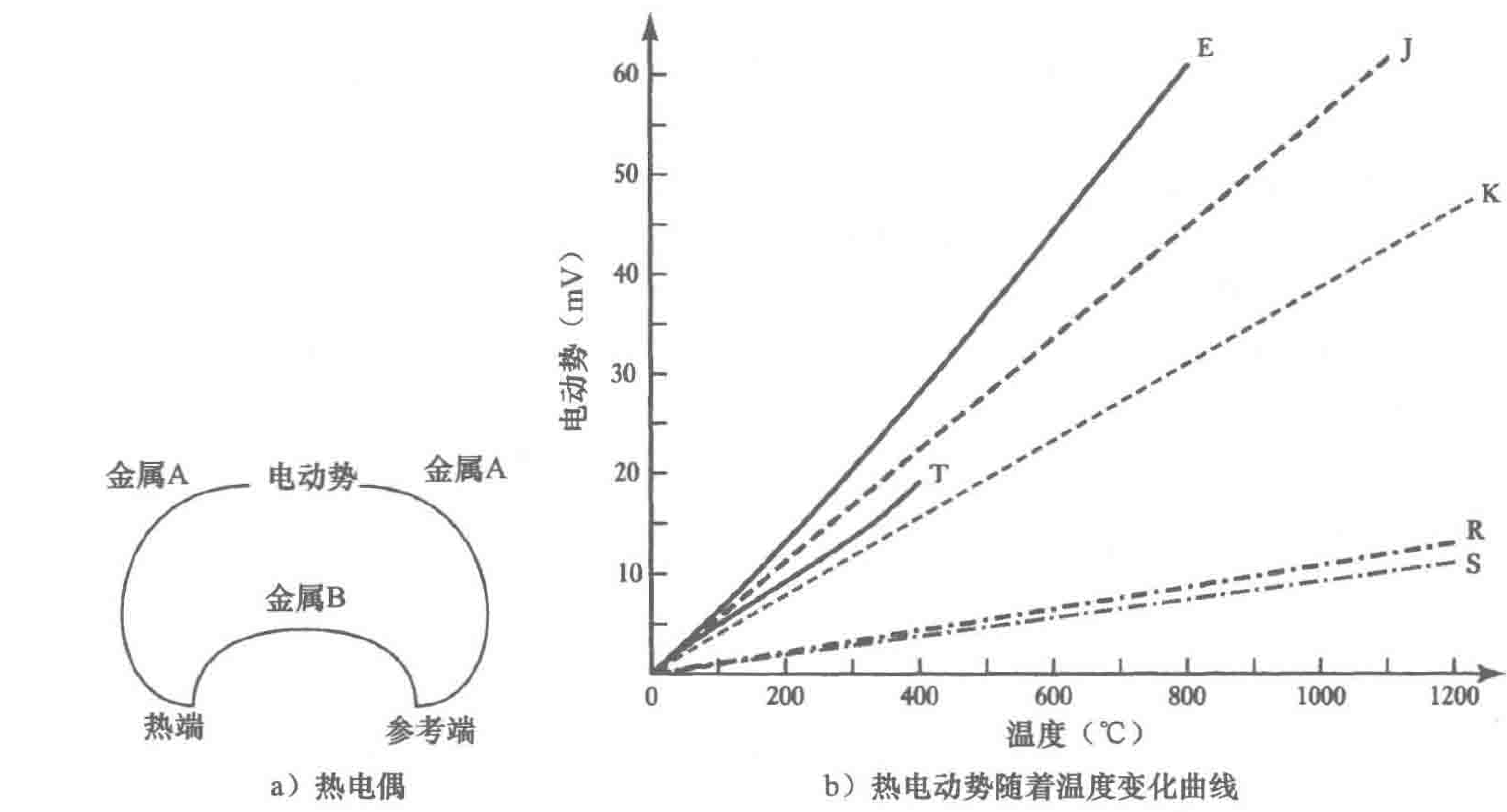


图 2-50 热电偶以及几个常用的热电偶的电动势随温度变化曲线

表 2-1 热电偶

分度号	材料	范围（℃）	（μV/℃）
B	铂铑（铑含量 30%）/铂铑（铑含量 6%）	0 ~ 1800	3
E	镍铬/康铜	-200 ~ 1000	63
J	铁/康铜	-200 ~ 900	53
K	镍铬/铝镍	-200 ~ 1300	41
N	镍铬硅/镍硅	-200 ~ 1300	28
R	铂/铂铑（铑含量 13%）	0 ~ 1400	6
S	铂/铂铑（铑含量 10%）	0 ~ 1400	6
T	铜/康铜	-200 ~ 400	43

59

一个热电偶电路中间可以有其他的金属，在同一温度下它们不会对热电偶连接端产生的热电动势产生影响，这就是中间导体定律。

热电偶参考端的温度可以不是 0℃。然而标准表假设参考端是 0℃，因此在使用表前需要有一个修正。这个修正就要使用**中间温度定律**，即

$$E_{t,0} = E_{t,I} + E_{I,0}$$

热电偶热端温度 t、冷端（参考端）温度为 0℃时的电动势 $E_{t,0}$等于两接点温度分别为 t、I 和 I、0℃时所产生的电动势 $E_{t,I}$、$E_{I,0}$之和。为了阐述这一点，考虑冷端为 20℃的 E 型热电偶的温度测量情况，它在 200℃的热电动势会是多大？下面是来自标准表中的数据：

温度（℃）	0	20	200
电动势（mV）	0	1.192	13.419

使用中间温度定律：

$$E_{200,20} = E_{200,0} - E_{20,0} = 13.419 - 1.192 = 12.227\text{mV}$$

注意：这不是表中给出的参考端温度为 0℃时热端为 180℃时的电动势 11.949mV。

为了保持一端的温度为 0℃，即需要浸入冰和水的混合物中，但通常不是很方便。通过补偿电路，我们可以得到一个随冷端温度变化而变化的电动势。当该电动势与热电偶的电动势相

加时，就得到了一个混合电动势。其数值等于热电偶冷端为0℃时得到的热端电动势（见图2-51）。补偿电动势可以由电阻温度计元件的压降来提供。

金属基热电偶E、J、K和T型相对便宜，但是性能会随着时间变差。它们通常的精度大约为±1%～3%。贵金属热电偶，比如说R型，更贵但是更稳定、寿命更长。它们的精度为±1%数量级或者更好。

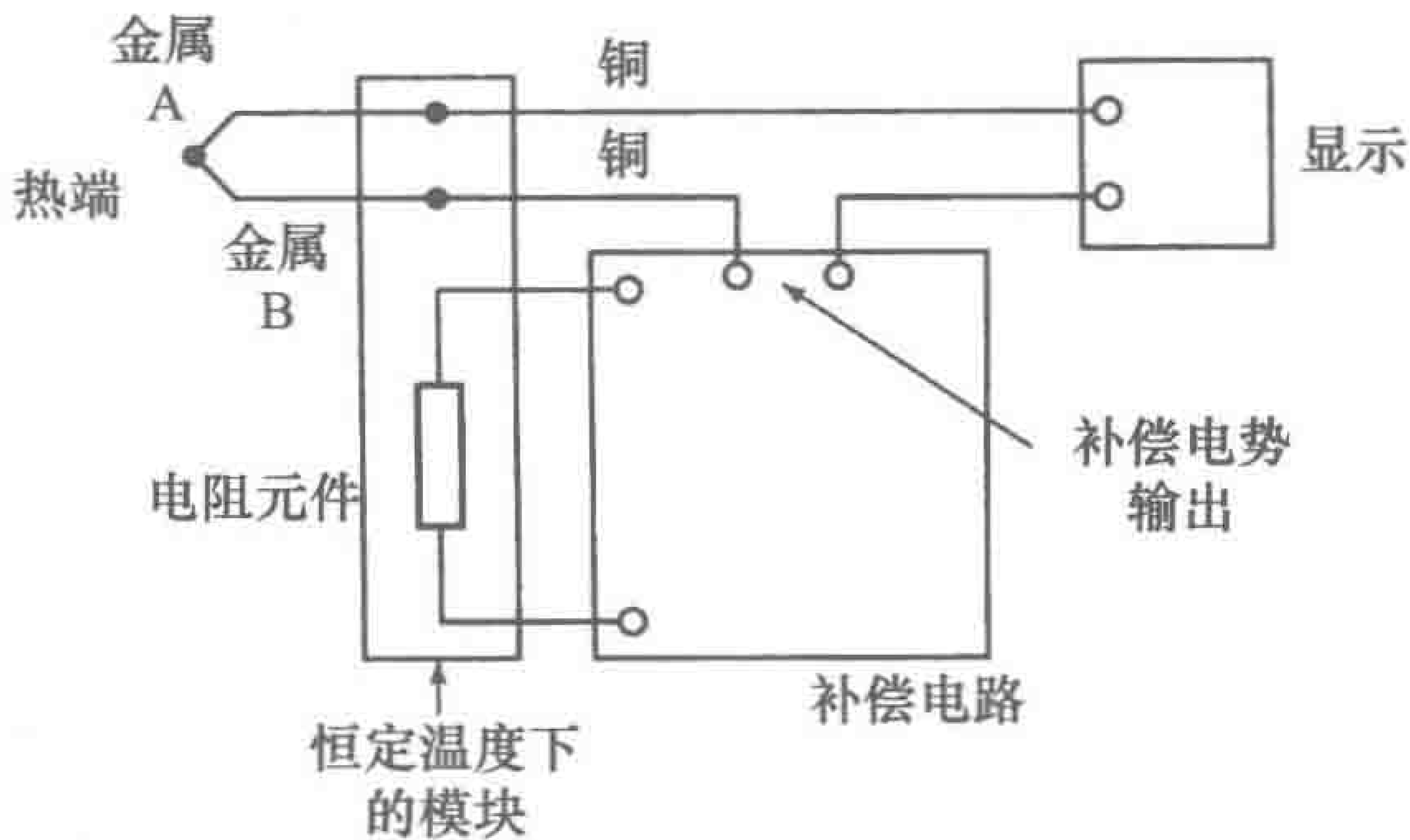


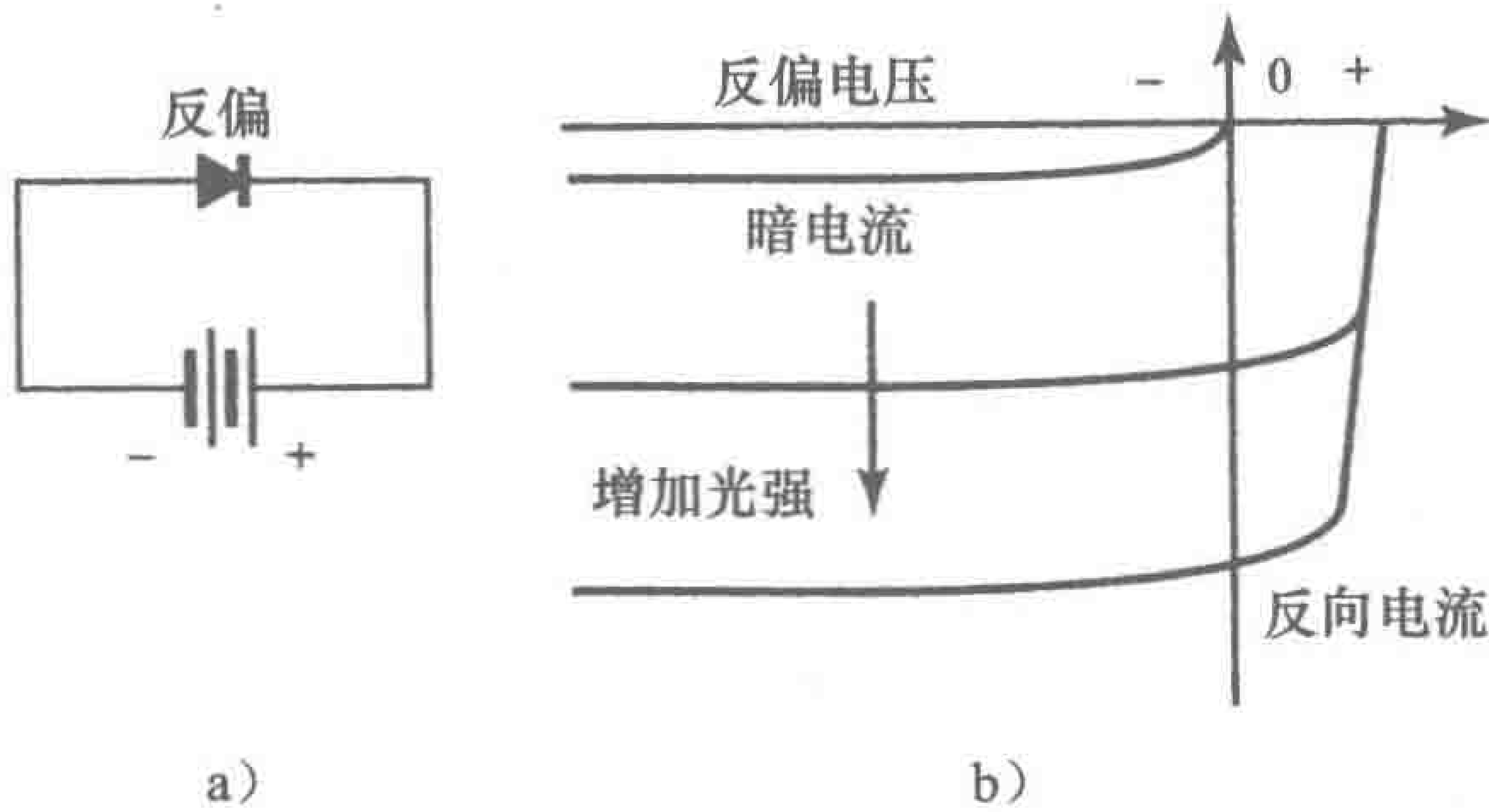
图2-51　冷端补偿

热电偶通常装在保护套上进行机械和化学防护。保护套的类型与热电偶工作温度有关。有时保护套用导热性很好且绝缘性很好的矿物包起来。没有保护套的热电偶响应时间非常快。如果用了很大的保护套，则响应时间会达到几秒。有些情况下，将一组热电偶串联起来，这样就可有10个或者更多的热电偶热端感应温度。产生的感应电动势加在一起，这样的使用方式称为**热电堆**（thermopile）。

60

2.10　光敏传感器

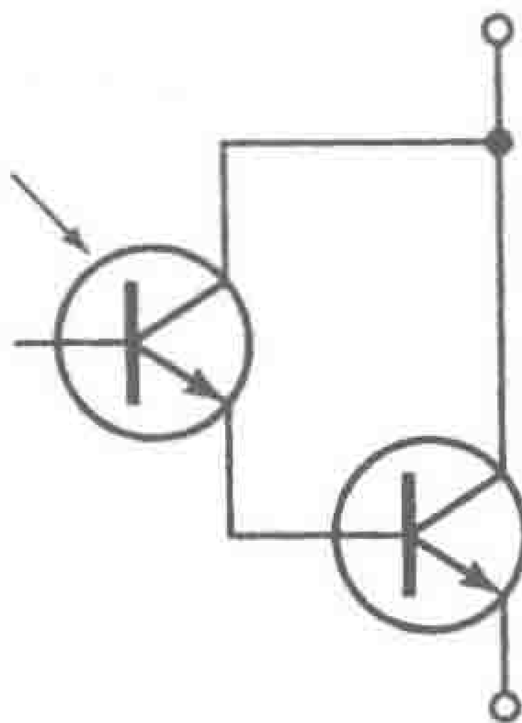
光电二极管（photodiode）是半导体结型二极管（见9.3.1节关于二极管的讨论），它反向连入电路，所以给出很大的电阻（见图2-52a）。当没有入射光时，反向电流可以忽略，称为暗电流。当光射入结时，产生的额外空穴电子增加反向电流，这样二极管电阻下降（见图2-52b）。反向电流几乎与光强成正比。例如，当没有光，且反向电压为3V时，电流为25μA，当照射了25 000lumens/m^{2}时，电流升到了375μA。没有光时二极管的电阻为$3/(25\times10^{-6})=120\text{k}\Omega$，有光时为$3/(375\times10^{-6})=8\text{k}\Omega$。因此，光电二极管可以用做一个由入射光控制的可变电阻器件。光电二极管对光的响应速度很快。



a）　b）

图2-52　光电二极管

光电晶体管（phototransistor，见9.3.3节关于晶体管的讨论）有一个光敏的集电极-基极p-n结。当没有光照时，集电极到发射极的电流很小。当有入射光时，基极产生的电流与光强成正比。这使得集电极电流能用于测量光强。光电晶体管通常可以与传统的晶体管连在一起形成达林顿复合管（见图2-53）。由于这样的组合会有更高的电流增益，所以器件会对给定的光强给出更大的集电极电流。


图2-53　光电达林顿管

光敏电阻（photoresistor）的阻值取决于照在上面的光强，光强增

加，电阻呈线性减小。硫化镉（cadmium sulphide）对波长小于 515nm 的光最敏感，硒化镉（cadmium selenide）光敏电阻则对波长小于 700nm 的光最敏感。

61

为了检测光强在某一空间的变化，我们通常需要在一个很小的空间里放置一排光敏传感器。一个例子是数码摄像机把捕捉的图像转为数字形式。为了这个目的，经常使用一个**电荷耦合装置**（Charge-Coupled Device，CCD）。一个 CCD 是由很多称为像素的光敏元件组成的光敏排列。这些元件基本上是 p 型硅层，通过耗尽层与 n 型硅层分隔开。当暴露在光中时，元件就会被充电，这些充电电荷通过电子电路变为 8 位的数字信号。在拍照时，数码摄像机的电子电路就给光敏元件放电，激活电动机械快门使这些元件在图像下曝光，然后读取每个元件的 8 位电荷值，这样就捕捉到了图像。因为 pn 元件无法分辨颜色，而我们需要的是彩色图像，所以光在射到元件之前先通过一个颜色滤镜矩阵，这样就能让绿色的光落在一些元件上，蓝色和红色分别落在其他元件上面。这样，考虑从不同元件获取的输出值，一幅彩色图像就产生了。

2.11　传感器的选择

在选择某个特殊应用的传感器时，需要考虑以下一些因素。

1）需要测量的性质。比如说，需要测量的变量、它的标称值、测量值的范围、精度要求、测量速度、可靠性要求、需要在哪种环境下进行测量。

2）传感器的输出性质。为了能给出合适的输出值，这个决定了信号调理的要求。

3）然后就可以确定可能的传感器，同时要考虑它们的测量范围、精度、线性度、响应速度、可靠性、可维护性、寿命、电源要求、坚固性、有效性，成本等因素。

传感器的选择不能和系统所要求的信号处理后的输出形式分离，因此需要传感器和信号处理的配合。

为了说明以上的问题，考虑测量容器中腐蚀酸液的液位的传感器的选择。液位能在一个直径为 1m 的圆柱形容器从0 ~ 2m 变化。空容器的重量为 100kg。最小的能够检测的液位的变化为 10cm。酸的密度为 1050kg/m^3。传感器的输出是电信号量。

由于酸的腐蚀性，所以间接测量液位比较合适。因此可以用一个称重元件或者很多称重元件去检测容器的重量。这样的元件会给出电量输出。液体的重量从空的时候重为 0，到满的时候重为 $1050 \times 2 \times \pi(1/4) \times 9.8 \approx 16.2$kN。再加上空容器的质量，这样重量的变化为从 1 ~ 17kN。分辨率为 10cm，也就是说，重量变化为 $0.10 \times 1050 \times \pi(1/4) \times 9.8 \approx 0.8$kN。如果用三个称重传感器支撑容器，这样每个需要称重为 0 ~ 6kN，分辨率约为 0.27kN。这样就可以查阅制造目录，看是否可以找到这样的称重元件。

62

2.12　通过开关输入数据

机械开关是由一对或者多对能够机械地闭合或打开的触点组成的，这样就能接通、断开电路。因此 0 和 1 的信号就能通过打开或者关上开关进行传输。**限位开关**的开或关是由一个物体的位移引起的，它用于指示动作的限位。

机械开关是按刀的数量和掷的数量决定的。**刀**（pole）是同一个切换动作能够形成的独立电路的数量，**掷**（throw）是每个刀的触点的数量。图 2-54a 显示的是一个单刀单掷（SPST）开关，图 2-54b 显示的是一个单刀双掷（SPDT）开关，图 2-54c 显示的是一个双刀双掷（DPDT）开关。

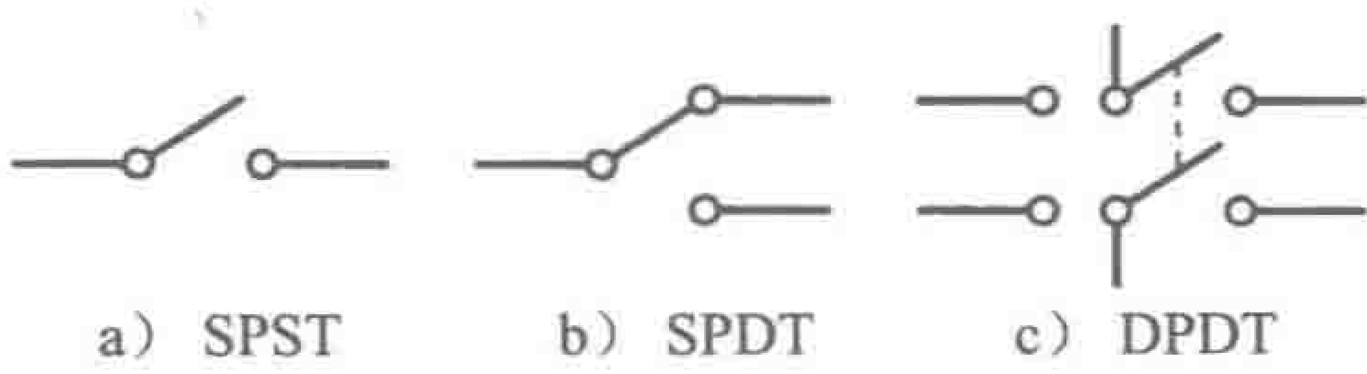
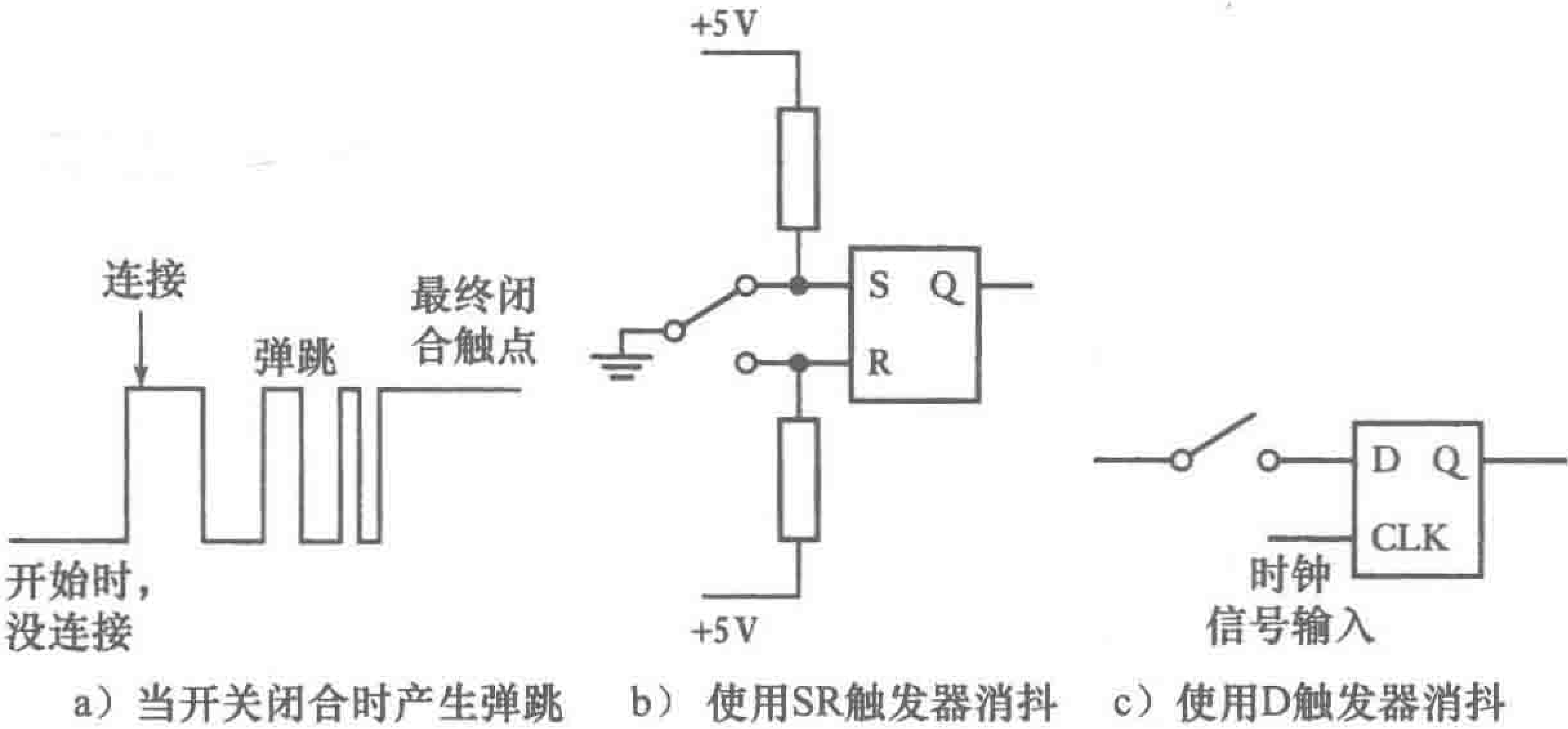


图 2-54　开关

2.12.1 消抖

机械开关的一个问题就是**开关抖动**问题。当一个开关被切换到触点闭合时，我们让一个触点移动到另一个触点。由于接触元件是弹性的，所以它碰到另一端的同时会发生抖动。它在大约20ms后达到稳定的闭合状态前，可能会跳动许多下（见图2-55a）。在弹跳过程中的每一次接触都可以认为是一次独立的接触，因此，对于微处理器来说，可能发生了两次或者更多次的开、关动作。类似地，当机械开关打开时，也会发生弹跳。为了解决这个问题可以采用软件和硬件的方法。

63

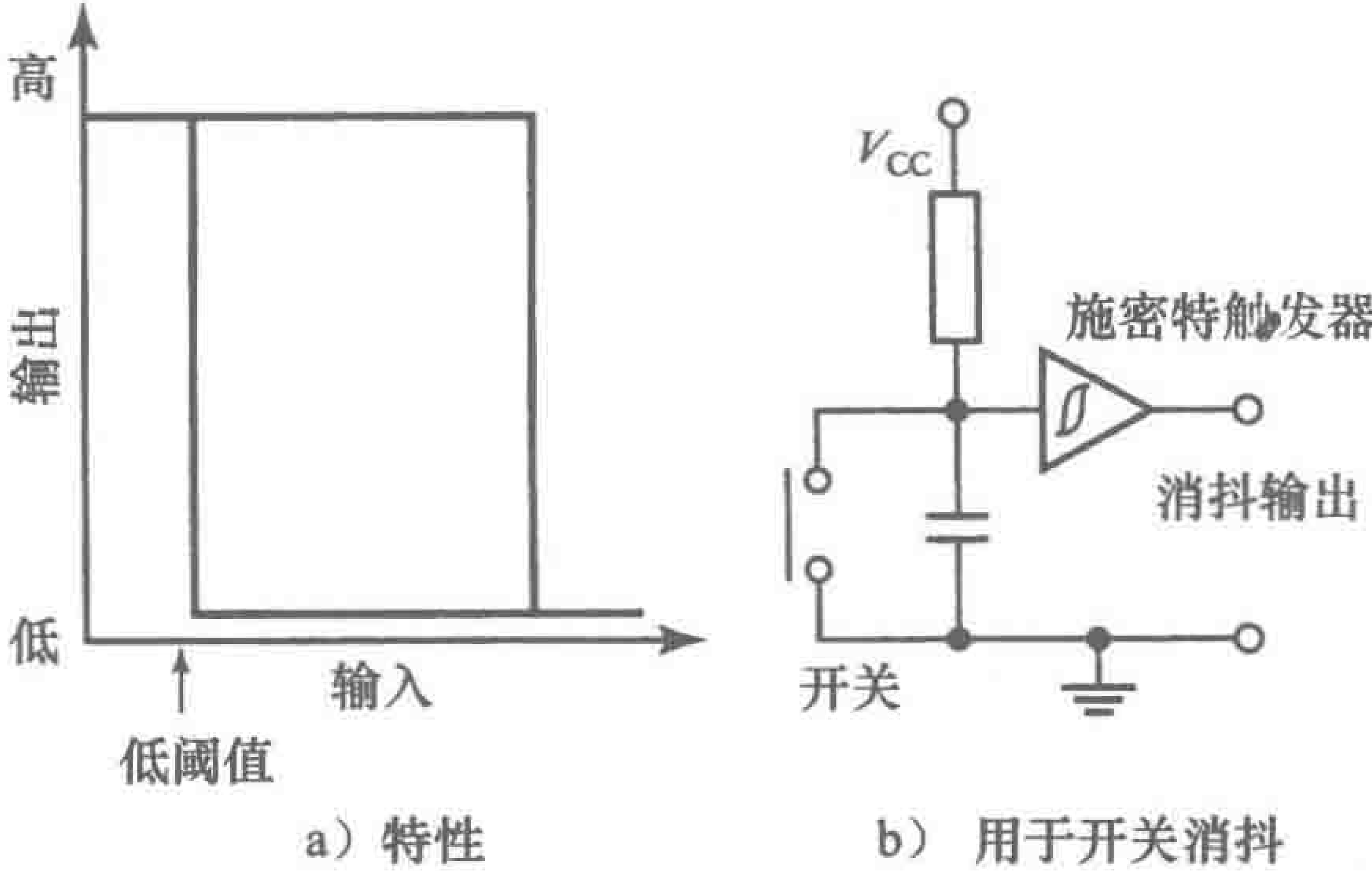


a）当开关闭合时产生弹跳 b）使用SR触发器消抖 c）使用D触发器消抖

图2-55 开关的消抖方法

用软件处理时，微处理器可以通过编程等待20s后再检测开关是否闭合上。当检查到弹跳已经停止且处于相同的闭合位置后，接下来的程序就可以开始执行了。硬件的解决方法是使用触发器。图2-55b显示的是一个使用SR触发器给单刀双掷开关消抖的电路（见5.4.1节）。如图2-55所示，我们把S设为0，R为1，输出为0，当开关被移到下面的位置时，S变为1，R变为0，输出为1。抖动使S从1到0到1再到0，等等，输出没有变化。这样的一个触发器能够由两个NOR门或者两个NAND门组成。一个单刀双掷开关也可以使用D触发器消抖（见5.4.4节）。图2-55c给出了电路。这样的触发器的输出只有当时钟信号变化时发生改变。因此可以选择一个比弹跳时间还长的时钟周期，比如说20ms，弹跳就可以忽略了。

还有一种选择消抖的方法是使用**施密特触发器**（Schmitt trigger）。这个器件的“迟滞”特性如图2-56a所示。当输入电压超过上切换阈值且给出低电平输出时，在输出电压转到高电平之前，输入电压需要降到低的阈值以下。相反，当输入电压处于低的切换阈值以下且给出高电平输出时，在输出变为低之前，输入需要升高到高阈值之上。这样的器件能够用于使缓慢变化的信号锐化：当信号通过切换阈值时，它会变为两个逻辑电平之间定义的一个直角边沿。如图2-56b所示的电路能够用于消抖，图中标注了施密特触发器的电路符号。当开关打开时，电容充电电压加到施密特触发器上，施密特触发器输入变高，并给出一个低的输出值。当开关关上时，电容放电非常快且第一次弹跳也使电容放电，因此施密特触发器就给出了高输出。连续的开关弹跳没有时间给电容再次充电到需要的阈值，这样之后的弹跳就不会触发施



a）特性 b）用于开关消抖

图2-56 施密特触发器

密特触发器。

2.12.2　键盘

键盘是一组开关，可以是计算机的键盘也可以是某些装置，比如说，微波炉的触摸屏。一个接触式的键盘通常都是如图 2-57a 所示的形式。按键推杆与弹簧相连，当按键释放时弹簧把按键又释放回原来的位置。一个典型薄膜开关（见图 2-57b）是由两片极薄的塑料薄膜组成，上面印制有导电层。这些层之间有层间隙。当薄膜的开关区域被按下时，顶层和底层接触在一起形成连接，当压力释放时打开。

64

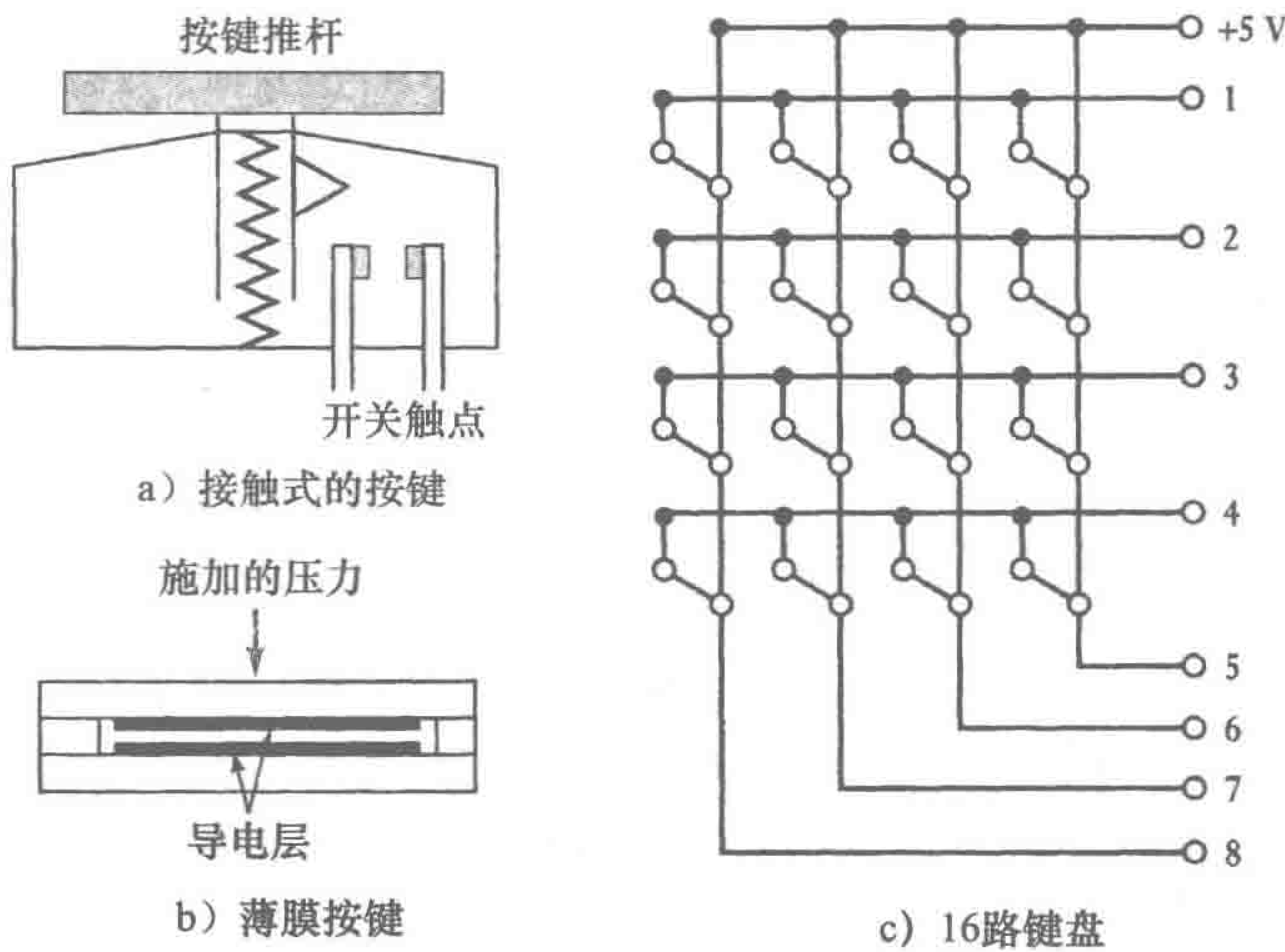


图 2-57　一个接触式键盘

在这样的阵列中，每个开关能够单独连接起来，且在关上的时候给出信号。一个更经济的方法是把它们连接在一个阵列中，其中不需要每个键有单独的输出，但是每个键可以给出唯一的行列组合。图 2-57c 显示了一个 16 路按键的键盘。

本章小结

传感器是一个能够产生与被测量有关系的信号的元件。**变送器**是一个对施加物理变化产生相关变化的元件。因此传感器就是变送器。然而，一个测量系统在把信号从一种形式转换为另一种形式时，除了传感器之外还可以使用变送器。

变送器的**范围**定义了输入能够变化的限度。**量程**是最大输入值减去最小输入值。**误差**是测量结果和真值的差。**精度**是测量结果可能出错的程度。**灵敏度**是单位输入对应的输出大小。**迟滞误差**是连续增长得到的值减去连续减小得到的值的差。**非线性误差**是假设按线性关系变化带来的误差。**可重复性/可再现性**是对同一输入的重复应用给出相同输出的能力。**稳定性**是固定常量输入给出相同输出的能力。**死区**是没有输出的输入范围。**分辨率**是产生可观察的输出变化的最小输入变化量。**响应时间**是阶跃输入后在输出达到某个输入的百分比的时间，比如 95%。时间常量是响应时间的 63.2%。**上升时间**是输出上升到稳定输出的某个百分比的时间。**建立时间**是输出达到稳态值某个百分比内所需的时间，比如 2%。

65

习题

2.1　解释以下变送器规格的含义：

(a) 压电式加速度计

非线性度：满量程 ±0.5%

(b) 电容式线性位移变送器

非线性及迟滞：满量程的 ±0.01%

(c) 电阻应变片力测量变送器

温度灵敏度：高于正常环境温度下满量程 ±1%

(d) 电容式流体压力变送器

精度：显示读数的 ±1%

(e) 热电偶

灵敏度：镍铬片/镍铝片热电偶：冷端0℃时0.039mv/℃

(f) 角速度测量陀螺仪变送器

可重复性：满量程 ±1%

(g) 电感式位移变送器

线性度：满量程 ±1%

(h) 称重传感器

考虑非线性性、迟滞及不可重复性的总误差：±1%。

2.2 康铜热电偶测温范围为0～200℃。电动势在0℃时为0mV，在100℃时为4.277mV，在200℃时为9.286mV。那么，假定电动势与温度整体为线性关系，在100℃时，电动势输出的非线性度误差的满量程百分比是多少？

2.3 热电偶从50℃的液体中取出并放入100℃液体中，从时间为0开始，得到以下电动势的值。求出95%热响应时间。

时间（s）	0	20	40	60	80	100	120
电动势（mV）	2.5	3.8	4.5	4.8	4.9	5.0	5.0

2.4 一个1kΩ的电位计，当负载为10kΩ，且处于最大位移的1/3时，其非线性度误差为多少？

2.5 对于一个阻值为50Ω，应变系数为2.1的电阻应变片，当应变为0.001时，其电阻变化为多少？

66

2.6 测量角位移时，使用增量轴编码器和绝对轴编码器得到的结果的主要区别在哪里？

2.7 使用轴编码器以50mm为半径测量线性位移时，若编码器每转一圈产生256个脉冲，则当位移为200mm时，其产生的脉冲数为多少？

2.8 一个旋转可变差动变压器（RVDT）的规格如下：

测量范围：±30度，满量程线性度误差 ±0.5%；±60°，满量程线性度误差 ±2.0%。

灵敏度：1.1（mv/v输入）/度。

阻抗：初级750Ω，次级2000Ω。

求：(a) 使用 ±60°范围的RVDT，读数为40°时的非线性度误差是多少？

(b) 输入电压为3V时，每度输出电压的变化是多少？

2.9 导电塑料薄膜式电位计与绕线式电位计相比有何优缺点？

2.10 某表面加装应变片的膜片压力传感器的规格如下：

测量范围：0～1400kPa，0～35 000kPa。

非线性误差：满量程 ±0.15%。

迟滞误差：满量程 ±0.05%。

那么测量范围在0～1400kPa，读数1000kPa时，由于非线性及迟滞误差所导致的总误差

为多少?

2.11　使用差压计监测开放容器的水位，可测量容器底部水位压力与大气的差压。若水位变化相对于差压计测量点在 0～2m，试求差压计所需量程是多少 kPa?

2.12　某铁康铜热电偶测温范围为 0～400℃，假定感应电动势与温度为线性关系，当读数为 100℃时，相对于满量程的非线性误差百分比是多少?（100℃时，电动势为 5.268mV；400℃时，电动势为 21.846mV。）

2.13　某铂电阻温度传感器，其阻值在 0℃时为 100Ω，100℃时为 138.5Ω，200℃时为 175.83Ω。求该传感器在读数为 100℃时的非线性误差值（假设在 0～200℃之间有非线性关系）。

2.14　某应变压力传感器的规格如下：

测量范围：2～70MPa，70kPa～1MPa。

激励源：10V 直流或者交流（方均根值）。

满量程输出：40mV。

非线性及迟滞误差：±0.5%。

适用温度范围：－54～＋120℃。

67

热零点漂移：0.030% 满量程输出/℃。

热漂移灵敏度：0.030% 满量程输出/℃。

试确定该传感器是否满足恒温 20℃环境下压力约为 100kPa，精度 ±5kPa 的使用要求?

2.15　一种用于测定容器水位的漂浮传感器，有一个总重 2.0kg、横截面积 $20cm^2$、长度 1.5m 的圆柱形漂浮物。它垂直漂浮在水中，向上力施加于附在它顶端末尾的梁上。由漂浮物作用在梁上的最小和最大推动力是多少呢? 建议通过一种方法，在推动力作用下测量出梁的变形量。

2.16　设计一种可以用于锅炉控制系统的传感器，测量热油流经管道的速度，测量系统的输出是一种可以用于调节油泵速度的电气信号。这个系统必须能够连续运行和自动工作，并且很长时间内不需调整。

2.17　设计一种可以用于控制系统的传感器来确定两个容器中液体的水位差。输出为控制系统提供一种电信号。

2.18　设计一种可以用于控制轧制板厚度系统的传感器，当轧制板从转轴中出来时测量它的厚度。金属片是连续运动的，测量必须快以确保校正行为能够很快地进行。这个测量系统必须提供一个电信号。

68

第3章　信号调理

本章目的

通过本章的学习，读者将能够：

- 解释信号调理的要求。
- 解释如何使用运算放大器。
- 解释保护和滤波的要求。
- 解释惠斯通电桥的原理，特别是它是如何与应变片一起工作的。
- 解释脉冲调制的原理。
- 解释接地回路和干扰会引起什么问题，并对这些问题提出可行的解决方法。
- 说明电气元件间最大功率传输的条件。

3.1　信号调理简介

通常，从测量系统的传感器输出的信号要通过某些处理后才能适合下一阶段的操作。比如说，信号可能太微弱而需要放大、信号包含需去除的干扰信号、信号是非线性的而需要线性化，信号是模拟量要转成数字量、数字量要转成模拟量、信号中的电阻变化要转化成电流变化、电压变化要转化成合适大小的电流变化等。所有这些转化都涉及**信号调理**。比如，一个从热电偶输出的电压只有几毫伏，可以用信号调理模块把它转换成合适大小的电流信号、提供抗噪声干扰、线性化和冷端温度补偿（即当热电偶的冷端不是0℃时的补偿）。

第4章将会继续讨论涉及数字信号的信号调理。

3.1.1　信号调理流程

下面是一些在信号调理中会使用的流程：

1）保护下一个元件（比如微处理器）不受高电流或电压的损害。因此微处理器需要串联限流电阻、需要在电流过大时能断开电路的保险丝、需要极性保护和限压回路（详见3.3节）。

2）把信号转换成合适的形式。这可以认为是把信号转化为直流电流和电压。比如，电阻应变片电阻的变化可以转换成电压的变化。这可通过惠斯通电桥和不平衡电压（详见3.5节）来转换。这表示可以把信号转换成数字量或模拟量（详见4.3节关于数模和模数转换器）。

3）使得信号处于合适的电平。热电偶输出的信号可能只有几毫伏，如果要把该信号输入到模数转换器来为微处理器提供输入，就需要把信号放大为几伏而不是几毫伏。运算放大器被广泛地用来放大信号（详见3.2节）。

4）消除或减少噪声。举例来说，滤波器可以用来消除信号中的主要噪声（详见3.4节）。

5）信号处理。比如把某些变量转换成线性的。这些信号可能来自于一些像流量计这样的传感器，它们输出的信号是非线性的，所以需要信号调理器来使信号在输入下一个元件前变成线性的（详见3.2.6节）。

下面各节概述一些信号调理中需要用到的元件。

3.2 运算放大器

从本质上来看，放大器可以认为是一个有输入和输出的系统（见图3-1），当输出和输入电压都是相对于地电压测量时，放大器的**电压增益**就是输出和输入电压的比值。放大器的**输入阻抗**定义为输入电压除以输入电流，**输出阻抗**则是输出电压除以输出电流。

很多信号调理模块的主要元件就是**运算放大器**。运算放大器是一种高增益的直流放大器，增益一般能达到100 000倍的数量级或更大，它可以组成硅片上的集成电路。运算放大器有两个输入端，分别是反相输入端（-）和同相输入端（+）。它的输出取决于这些输入之间的连接。运算放大器上还有其他的输入端，称为负电源端、正电源端和两个调零输入端，它们可以用来修正放大器的非理想行为（详见3.2.8节）。图3-2展示了741型运算放大器的管脚定义。

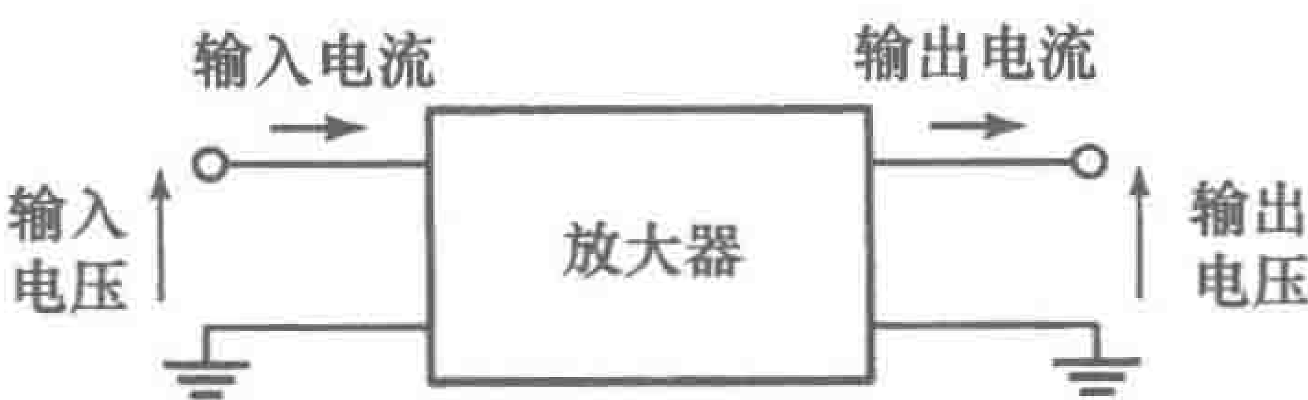


图3-1 放大器

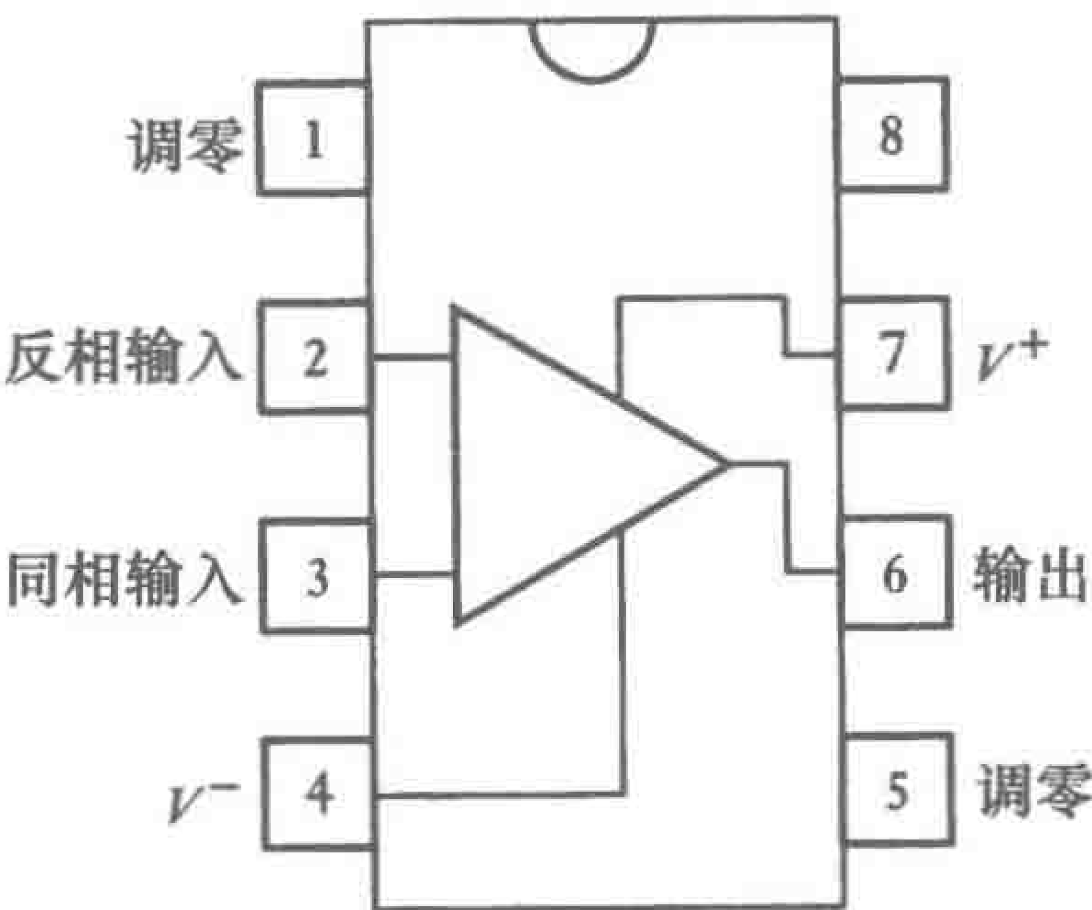


图3-2 741运算放大器的管脚连接

一个理想的运算放大器模型是一个增益和输入阻抗无穷大、输出阻抗为零的放大器，即输出电压独立于负载。

70

下面将列出几种信号调理器中可能用到的运算放大器电路。

3.2.1 反相放大器

图3-3展示了放大器作为**反相放大器**使用时的连接。放大器反相输入端串联着电阻R_1，同相输入端接地。从输出端引出了一条反馈支路，串联上电阻R_2后回到了反相输入端。运算放大器的电压增益在100 000倍左右，而输出电压的变化一般都限制在±10V左右。所以输入电压必须在-0.0001～+0.0001V之间。输入电压几乎为0，所以实际上X点处近似为地的电势。因此这又叫作**虚地**。R_1两端的电势差是$(V_{in}-V_X)$。因此，由于理想运算放大器的增益无穷大且$V_X=0$，所以输入电势V_{in}可以认为全落在电阻R_1上。即

$$V_{in}=I_1R_1$$

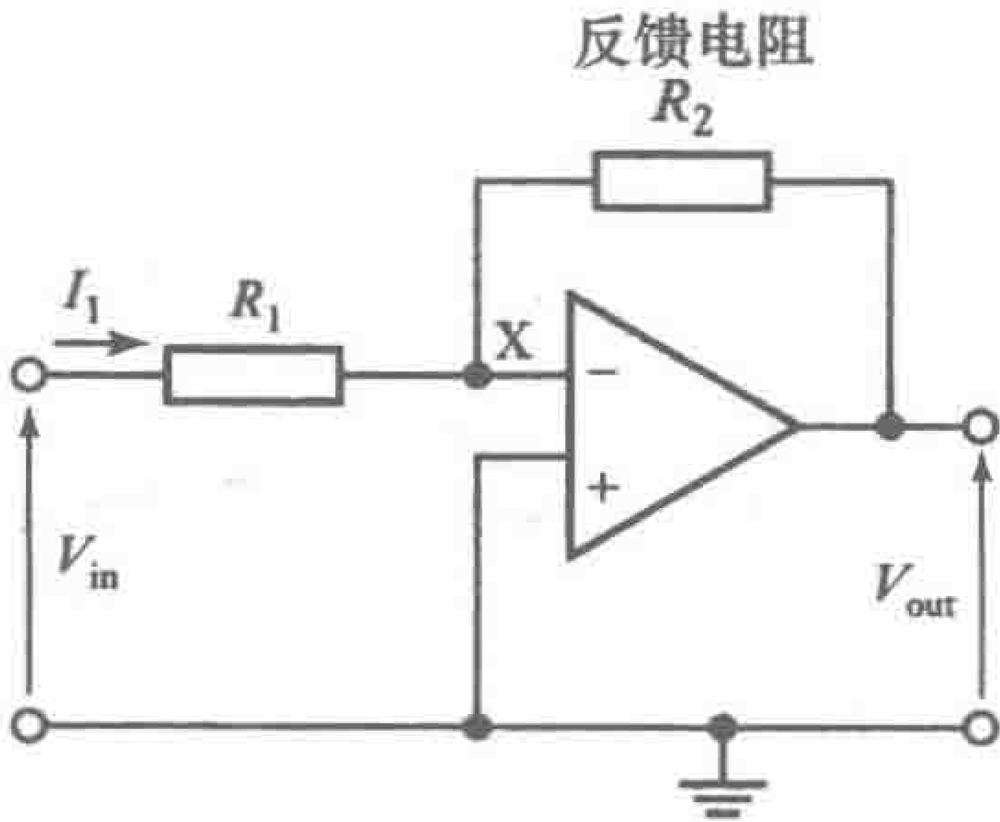


图3-3 反相放大器

运算放大器输入端之间有很高的阻抗。比如741型就有大约2MΩ。因此几乎没有电流从X点流进放大器。因为理想的运算放大器输入阻抗无穷大，所以没有电流从X点流入。因此流过R_1的电流I_1必然流过R_2。R_2两端的电势差为(V_X-V_{out})，又由于理想运算放大器V_X为0，所以R_2两端的电势差就是$-V_{out}$。即

$$-V_{out}=I_1R_2$$

上面两个方程相除，得到

$$\text{电路电压增益} = \frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1}$$

所以电路的电压增益只取决于 R_2 和 R_1 的相对值。负号表示输出是反相的，即相对输入落后了180°的相位。

举个例子来说明，一个反相运算放大器电路的反相输入端串联了一个 1MΩ 电阻而反馈电阻为 10MΩ，这个电路的电压增益是多少?

71

$$\text{电路电压增益} = \frac{V_{out}}{V_{in}} = -\frac{R_2}{R_1} = -\frac{10}{1} = -10$$

作为一个反相放大器电路的使用例子，光电二极管（详见 2.10 节）作为传感器被广泛地使用，它能在受到光照时提供微电流。当反相放大器电路中使用了这样的传感器来给出一个电流到电压的转换器时，需要将光电二极管反偏接入电阻 R_1 的位置，这样才能使得输出作为微控制器的输入。

3.2.2　同相放大器

图 3-4a 展示了运算放大器作为同相放大器使用时的连接。输出可以认为来自 R_1 和 R_2 串联构成的分压电路电压。V_X 是输出电压的 $R_1/(R_1+R_2)$ 倍，即

$$V_X = \frac{R_1}{R_1+R_2}V_{out}$$

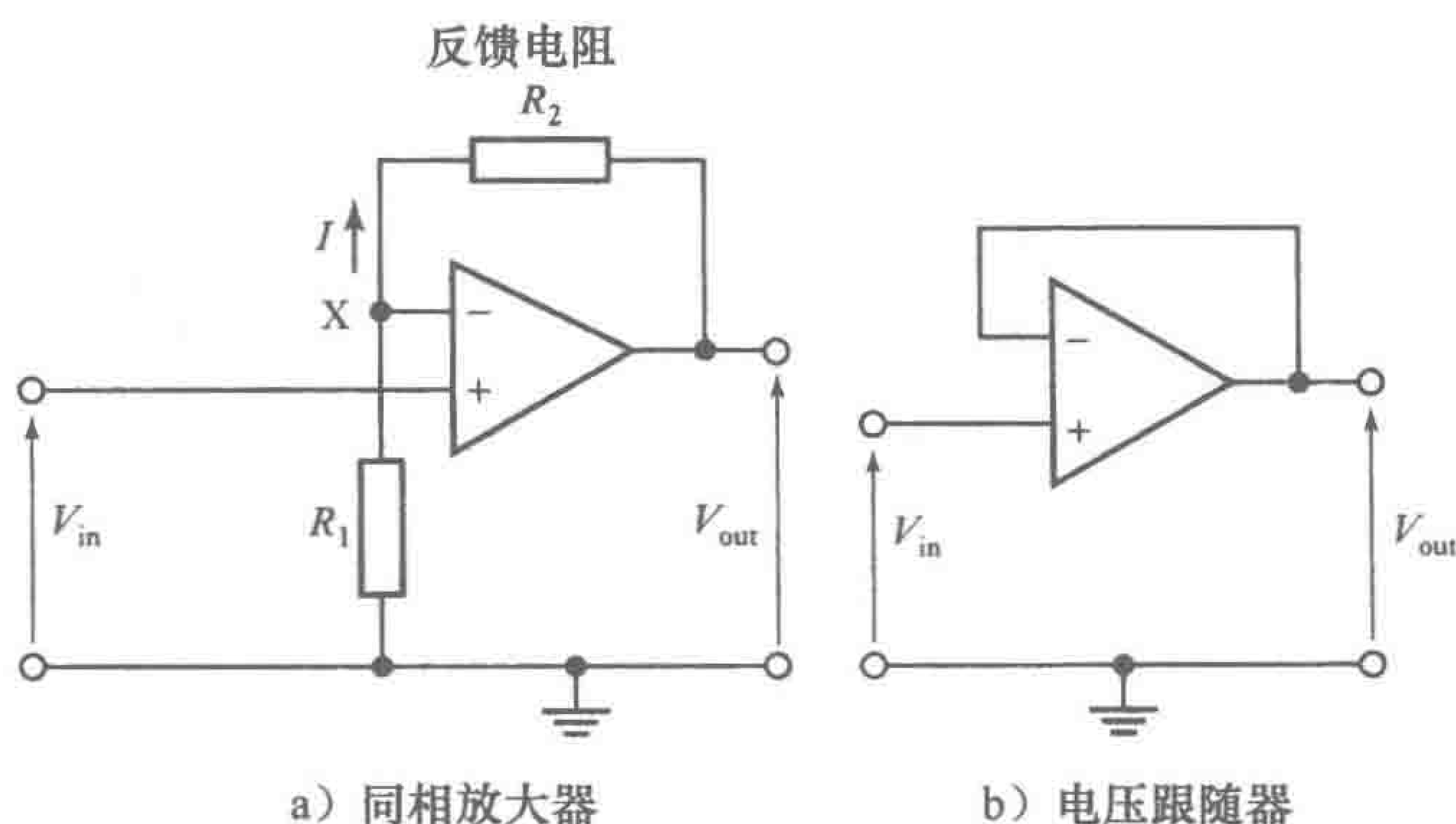


图 3-4　同相放大器和电压跟随器

既然实际上没有电流流经运算放大器的两个输入端，所以它们之间也就没有电势差。也就是说，对于理想的运算放大器一定有 $V_X = V_{in}$。所以，

$$\text{电路电压增益} = \frac{V_{out}}{V_{in}} = \frac{R_1+R_2}{R_1} = 1 + \frac{R_2}{R_1}$$

当反馈回路短接时，即 $R_2 = 0$ 时，就成为这类放大器中的一种特殊形式。这时电压增益为1。电路的输入电阻很大，一般能有 2MΩ。而输出电阻，即输出端和地线之间的电阻，比输入电阻小很多，一般约为 75Ω。因此后一级电路电阻相对较小，该放大器不易使负载对该电路产生影响。这样的放大器称为**电压跟随器**，图 3-4b 展示了它的基本电路。

3.2.3　加法放大器

图 3-5 展示了加法放大器的电路。和反相放大器（详见 3.2.1 节）一样，X 点是虚地。因此流入 X 点的电流总和必然等于流出的电流。即

72

$$I = I_A + I_B + I_C$$

其中 $I_A = V_A/R_A$，$I_B = V_B/R_B$，$I_C = V_C/R_C$。而且，电流 I 必然流经反馈电阻 R_2。R_2 两端的电势

差为（$V_X - V_{out}$）。又由于 V_X 可以认为是0，所以电势差就是 $-V_{out}$，故 $I = -V_{out}/R_2$。因此，

$$-\frac{V_{out}}{R_2} = \frac{V_A}{R_A} + \frac{V_B}{R_B} + \frac{V_C}{R_C}$$

输出是各输入的比例之和，即

$$V_{out} = -\left(\frac{R_2}{R_A}V_A + \frac{R_2}{R_B}V_B + \frac{R_2}{R_C}V_C\right)$$

如果 $R_A = R_B = R_C = R_1$，那么

$$V_{out} = -\frac{R_2}{R_1}(V_A + V_B + V_C)$$

举例来说，要设计一个来自三个传感器输入电压的均值作为输出电压的电路。假设允许输出为反相的，那么图3-5展示的电路就能派上用场。三个输入中每个都乘以1/3来作为平均的输出。因此每个输入信号都为电路电压增益的1/3就能满足要求了。所以，如果反馈电阻为4kΩ，则每个输入端的电阻就应为12kΩ。

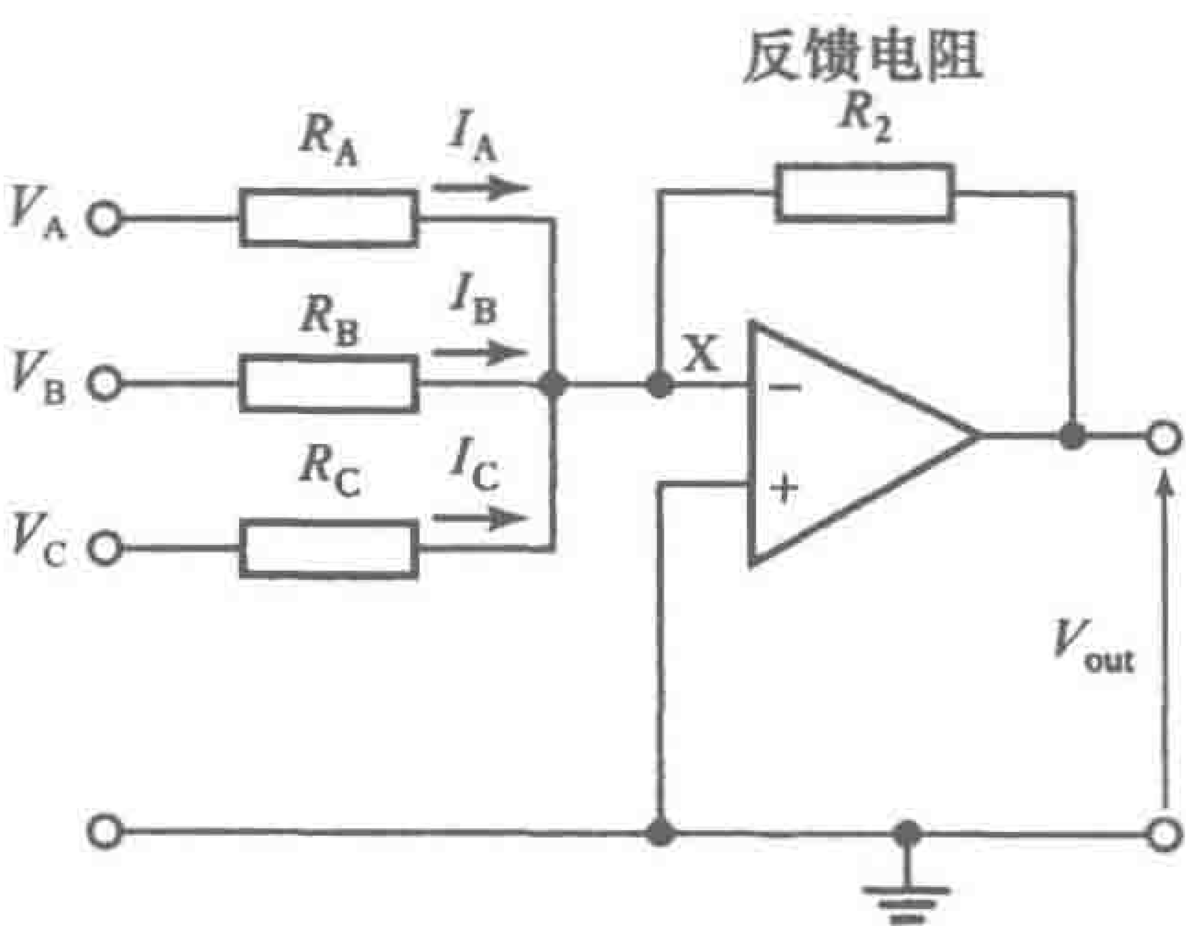


图3-5 求和放大器

3.2.4 积分和微分放大器

如图3-6a所示，考虑一个通过电容反馈的反相运算放大器电路。

电流是电荷 q 的变化率，又由于对于电容有 $q = Cv$，其中 v 就是电容两端的电压，所以流经电容的电流 $i = dq/dt = Cdv/dt$。C 两端的电势差为（$v_X - v_{out}$），由于虚地 v_X 几乎为0，所以电势差为 $-v_{out}$。因此流经电容的电流为 $-Cdv_{out}/dt$。这也同样是流经输入电阻 R 的电流。所以

$$\frac{v_{in}}{R} = -C\frac{dv_{out}}{dt}$$

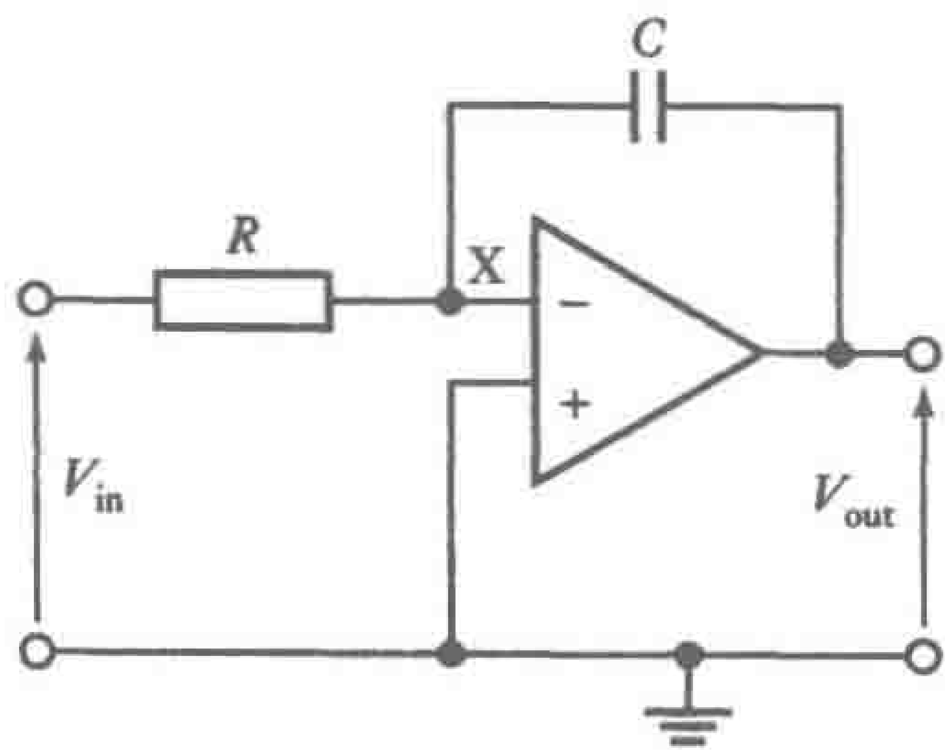
整理之后，为

$$dv_{out} = -\left(\frac{1}{RC}\right)v_{in}dt$$

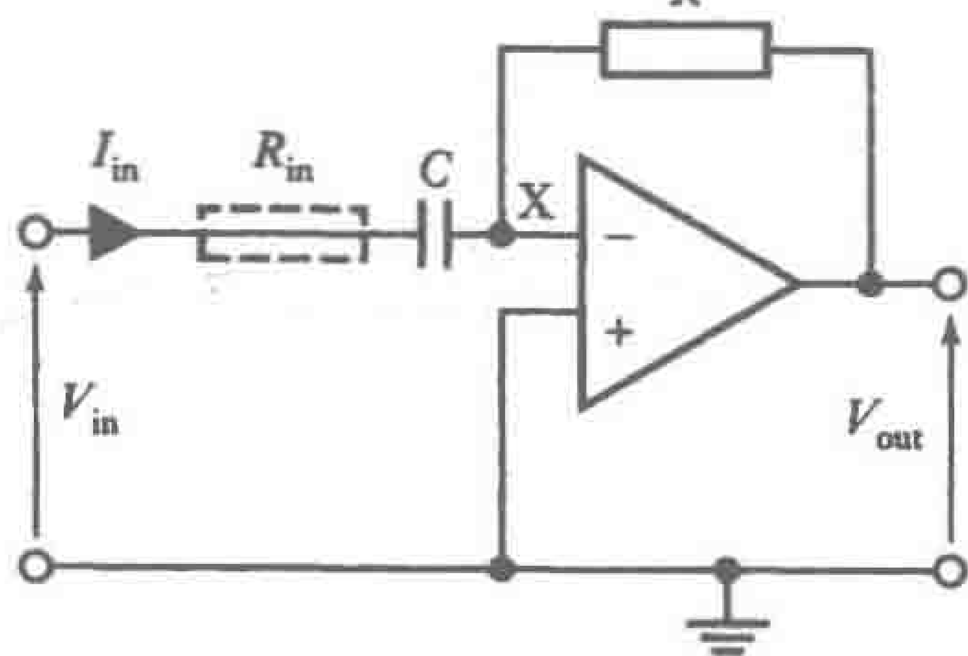
对上式两端同时积分，得

$$v_{out}(t_2) - v_{out}(t_1) = -\frac{1}{RC}\int_{t_1}^{t_2} v_{in}dt$$

$v_{out}(t_2)$ 是 t_2 时的输出电压，$v_{out}(t_1)$ 是 t_1 时的输出电压。输出和输入电压的积分（即输入电压随时间变化的曲线下的面积）成比例。



a）积分放大器



b）微分放大器

图3-6 积分放大器和微分放大器

如果积分放大器电路中电容和电阻互换，那么就是微分电路。图3-6b展示了这个电路。流入电容 C 的输入电流为 $dq/dt = Cdv/dt$。理想状态下流经运算放大器的电流，也就是流经反馈电阻 R 的电流为 $-v_{out}/R$，所以

$$\frac{v_{out}}{R} = -C\frac{dv_{in}}{dt}$$

$$v_{out} = -RC\frac{dv_{in}}{dt}$$

微分电路的高频易于发生稳定性和噪声问题。一种解决方式就是加入一个输入电阻 R_{in} 来

73

限制高频时的增益从而减少这些问题。

3.2.5　差分放大器

差分放大器是一种能放大两个输入电压差值的放大器（见图3-7）。既然几乎没有电流流经运算放大器两个输入端口间的大阻抗电阻，所以这里就没有电压降，两个输入点X有一样的电势。V_2 是电阻 R_1 和 R_2 串联后两端的电压。因此X点的电势 V_X 就是

$$\frac{V_X}{V_2}=\frac{R_2}{R_1+R_2}$$

74

流经反馈电阻的电流一定等于从 V_1 流过 R_1 的电流。所以

$$\frac{V_1-V_X}{R_1}=\frac{V_X-V_{out}}{R_2}$$

整理后为

$$\frac{V_{out}}{R_2}=V_X\left(\frac{1}{R_2}+\frac{1}{R_1}\right)-\frac{V_1}{R_1}$$

用之前的等式消去 V_X 后，得

$$V_{out}=\frac{R_2}{R_1}(V_2-V_1)$$

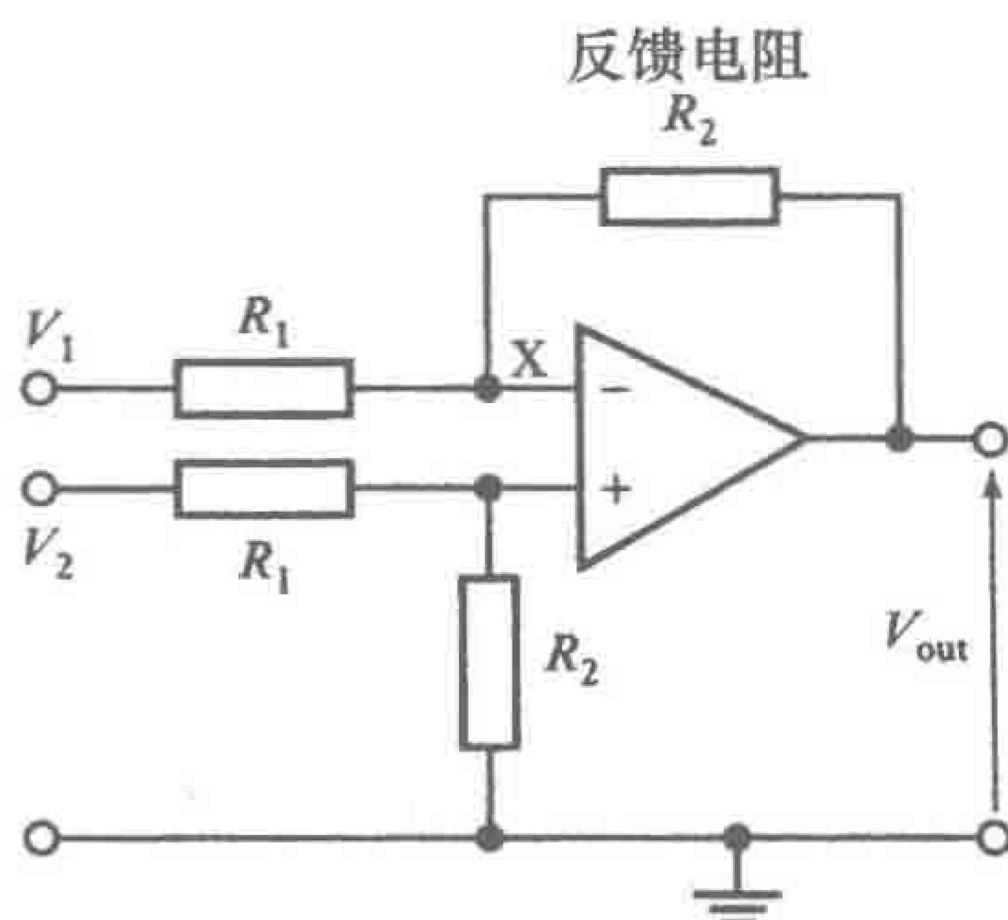


图3-7　差分放大器

因此输出为两个输入电压的差值。

为了用传感器说明上述电路的应用，图3-8展示了它如何和热电偶一起使用。电路放大了热电偶两端电动势间的电压差。比如说，当热电偶两端有10℃的温差时，如果这一温差可以使两端产生的电动势差值达到530μV，要想选择 R_1 和 R_2 的大小来使电路的输出为10mV，那么对于这一电路我们有：

$$V_{out}=\frac{R_2}{R_1}(V_2-V_1)$$

$$10\times10^{-3}=\frac{R_2}{R_1}\times530\times10^{-6}$$

所以 $R_2/R_1=18.9$。因此如果 R_1 为10kΩ的电阻，那么 R_2 就必须为189kΩ。

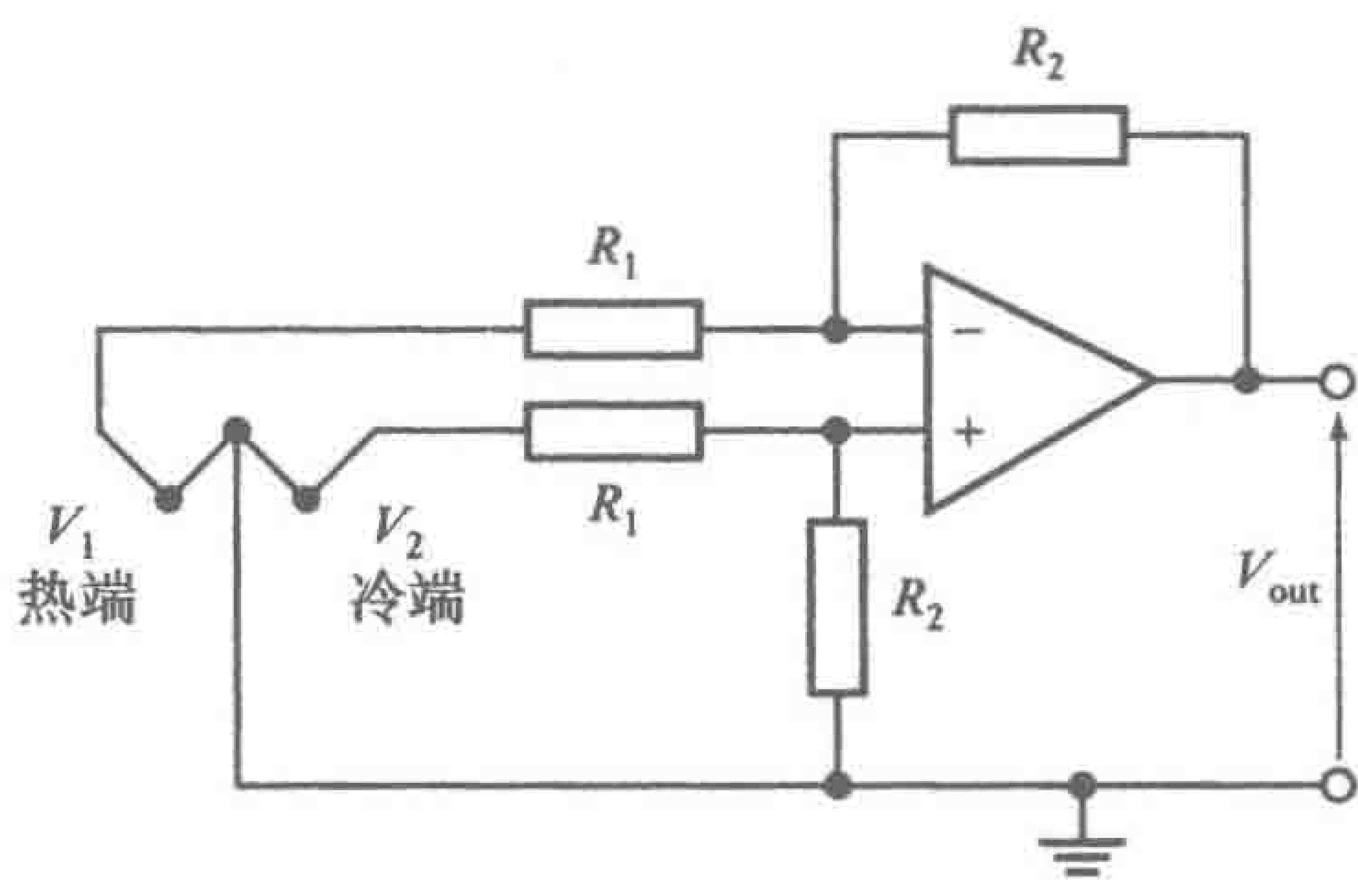


图3-8　带有热电偶的差分放大器

差分放大器也可以和惠斯通电桥（详见3.5节）一起使用，可能把电阻应变片放在电桥的臂中来放大电桥中一个或多个臂的电阻值改变而产生的不平衡电势差。当电桥平衡时，电桥的两个输出端电势相同，因此也没有电势差。比如输出端可能都是5V。所以差分放大器的输入端也都是5V。当电桥不再平衡时，比如

75

一端为5.01V另一端为4.99V，差分放大器的输入就为5.01V和4.99V。放大器会放大这一0.02V的电压差值。原来的5.00V信号对于两个输入端都是相同的，所以称为**共模电压**（common mode voltage）V_{CM}。在假设两个输入通道完全相等而运算放大器对它们的放大增益也是一样的前提下，放大器只放大两个信号的差值。而实际上这些条件都是不可能完全达到的，因此输出与两个输入的电压差也不是完美的正比关系。所以我们可把输出写为：

$$V_{out}=G_d\Delta V+G_{CM}V_{CM}$$

G_d 是电压差 ΔV 的增益，G_{CM} 是共模电压 V_{CM} 的增益。G_{CM} 的值越小，共模电压对输出的影响也就越小。一个运算放大器偏离理想状况的程度用**共模抑制比**（Common Mode Rejection Ratio，CMRR）来表示：

$$\mathrm{CMRR} = \frac{G_{\mathrm{d}}}{G_{\mathrm{CM}}}$$

为了将共模电压对输出的影响降到最小，就要求有一个高的 CMRR。CMRR 一般可以用分贝（decibel，dB）来表示。因此，在分贝尺度下，10 000 的 CMRR 就是 20lg10 000 = 80dB。运算放大器的典型 CMRR 一般在 80 ~ 100dB 之间。

常见的**仪表放大器**包含三个运算放大器（见图 3-9），而不是只有一个差分放大器，被集成在一个电路中。这样的电路被设计成有高输入阻抗，一般有 300MΩ，还有高电压增益和极佳的 CMRR（一般高于 100dB）。电路的第一部分包含放大器 A_1 和 A_2，一个被连接成反相放大器，另一个连接成同相放大器。放大器 A_3 是把 A_1 和 A_2 的输出作为输入的差分放大器。

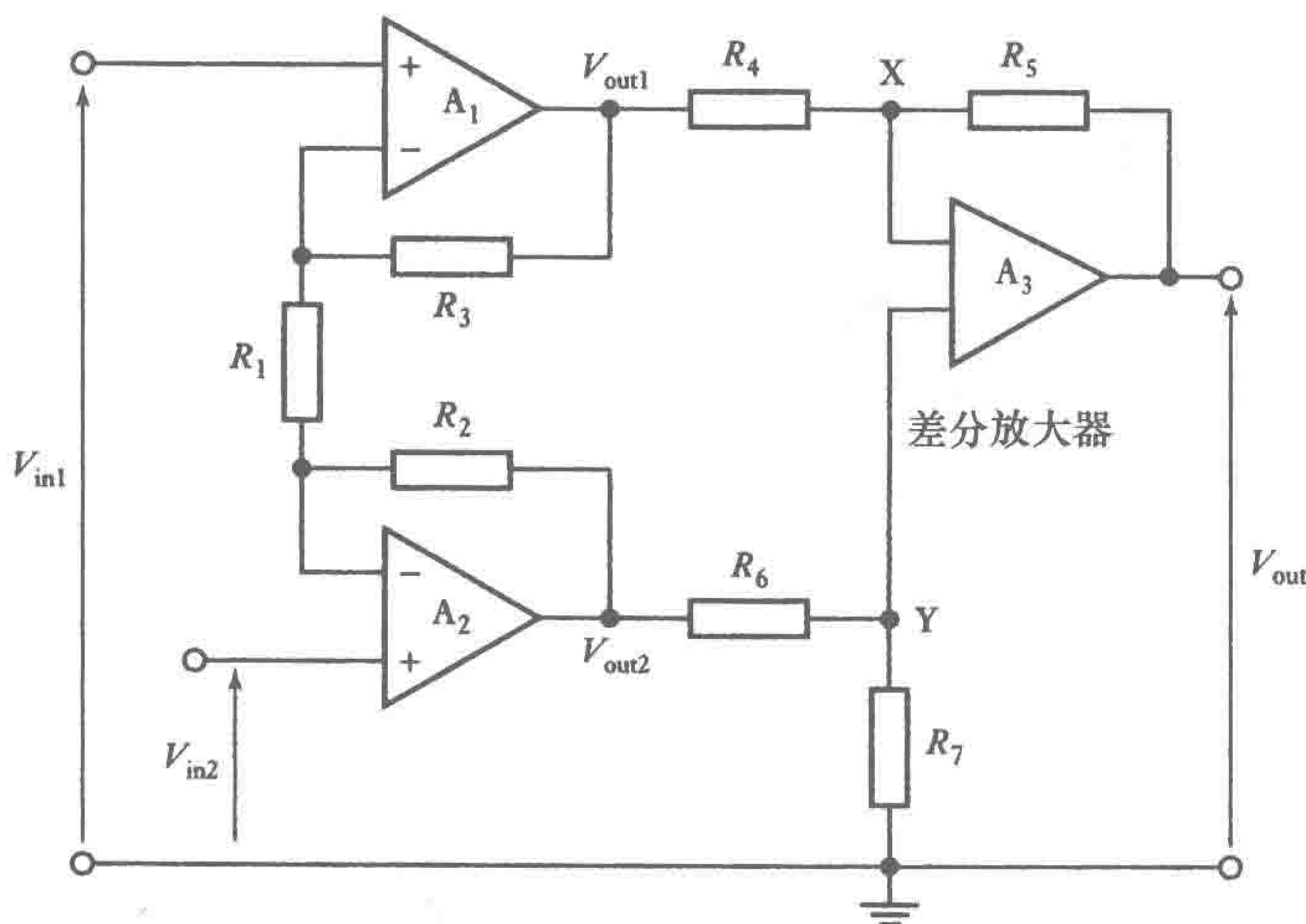


图 3-9　仪表放大器

76

因为几乎没有电流流经 A_3，所以流经 R_4 的电流和流经 R_5 的电流相等。因此

$$\frac{V_{\mathrm{out1}} - V_{\mathrm{X}}}{R_4} = \frac{V_{\mathrm{X}} - V_{\mathrm{out}}}{R_5}$$

A_3 的输入差值基本为零，所以 $V_{\mathrm{Y}} = V_{\mathrm{X}}$。因此上式就可写成

$$V_{\mathrm{out}} = \left(1 + \frac{R_5}{R_4}\right) V_{\mathrm{Y}} - \frac{R_5}{R_4} V_{\mathrm{out1}}$$

R_6 和 R_7 是电压 V_{out2} 的分压器，有

$$V_{\mathrm{Y}} = \frac{R_6}{R_6 + R_7} V_{\mathrm{out2}}$$

因此可以写成

$$V_{\mathrm{out}} = \frac{1 + \dfrac{R_5}{R_4}}{1 + \dfrac{R_7}{R_6}} V_{\mathrm{out2}} - \frac{R_5}{R_4} V_{\mathrm{out1}}$$

所以通过选择合适的电阻值，我们可以使两个差分放大器的输入的放大倍数相等。这需要

$$1 + \frac{R_5}{R_4} = \left(1 + \frac{R_7}{R_6}\right) \frac{R_5}{R_4}$$

因此 $R_4/R_5 = R_6/R_7$。

我们可以利用**叠加原理**（principle of superposition），也就是说，我们可以考虑把每个信号源产生的输出单独计算，然后加起来得到总的结果。放大器 A_1 的同相输入端有一个输入信号 V_{in1}，并以 $1+R_3/R_1$ 的增益放大这一输入。在它的反相输入端有来自 V_{in2} 的输入，并以 $-R_3/R_1$ 的增益放大。而同相输入的共模电压 V_{cm} 也被 A_1 放大了。因此 A_1 的输出就是

$$V_{out1}=\left(1+\frac{R_3}{R_1}\right)V_{in1}-\left(\frac{R_3}{R_1}\right)V_{in2}+\left(1+\frac{R_3}{R_1}\right)V_{cm}$$

类似地，放大器 A_2 有

$$V_{out2}=\left(1+\frac{R_2}{R_1}\right)V_{in2}-\left(\frac{R_2}{R_1}\right)V_{in1}+\left(1+\frac{R_2}{R_1}\right)V_{cm}$$

对 A_3 的差分输入为 $V_{out2}-V_{out1}$，所以

77

$$V_{out2}-V_{out1}=\left(1+\frac{R_3}{R_1}+\frac{R_2}{R_1}\right)V_{in2}-\left(1+\frac{R_2}{R_1}+\frac{R_3}{R_1}\right)V_{in1}+\left(\frac{R_2}{R_1}-\frac{R_3}{R_1}\right)V_{cm}$$

当 $R_2=R_3$ 时就消除了共模电压，因此

$$V_{out2}-V_{out1}=\left(1+\frac{2R_2}{R_1}\right)(V_{in2}-V_{in1})$$

总的增益就为 $(1+2R_2/R_1)$ 且通常通过改变 R_1 来设定这个值。

图 3-10 展示了管脚间的关系和一种利用这三种运算放大器所设计的通用的低成本仪表放大器（Burr-Brown INA114）的一些详细说明。增益通过在管脚 1 ~ 8 之间连接一个外部电阻 R_G 来设定，当 R_G 的单位为 kΩ 时，增益为 $1+50/R_G$。分子上的 50kΩ 来自两个内部反馈电阻之和。

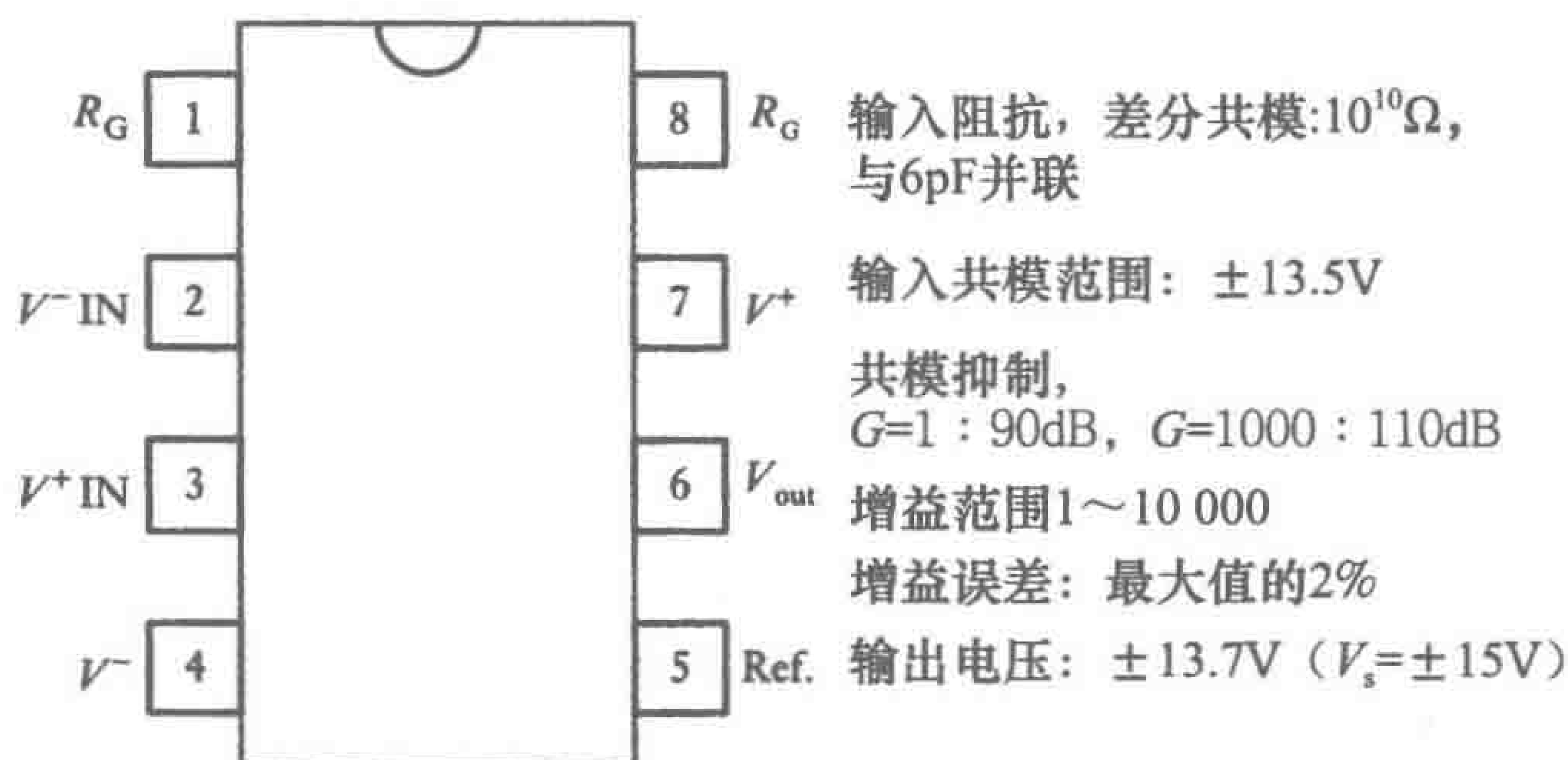


图 3-10　INA114

3.2.6　对数放大器

有些传感器的输出是非线性的。比如，热电偶的输出关于冷热端之间的温差就不是一个完全的线性函数。这类传感器的输出需要信号调理器来线性化。可以利用一种把输入/输出关系设计为非线性的运算放大器，这样当输入是非线性时，输出就是线性的。这需要靠在反馈回路上选择合适的元件来完成。

图 3-11 所展示的对数放大器就是这类信号调理器的一个例子。反馈回路包含了一个二极管（或者一个基极接地的晶体管）。二极管有非线性的特性。可以表示为 $V=C\ln I$，其中 C 为常量。那么，由于流经反馈回路的电流和流经输入电阻的电流相等，且二极管两端的电势差为 $-V_{out}$，所以有

$$V_{out}=-C\ln(V_{in}/R)=K\ln V_{in}$$

其中 K 为常量。然而，如果输入 V_{in} 是由一个以 t 为输入的传感

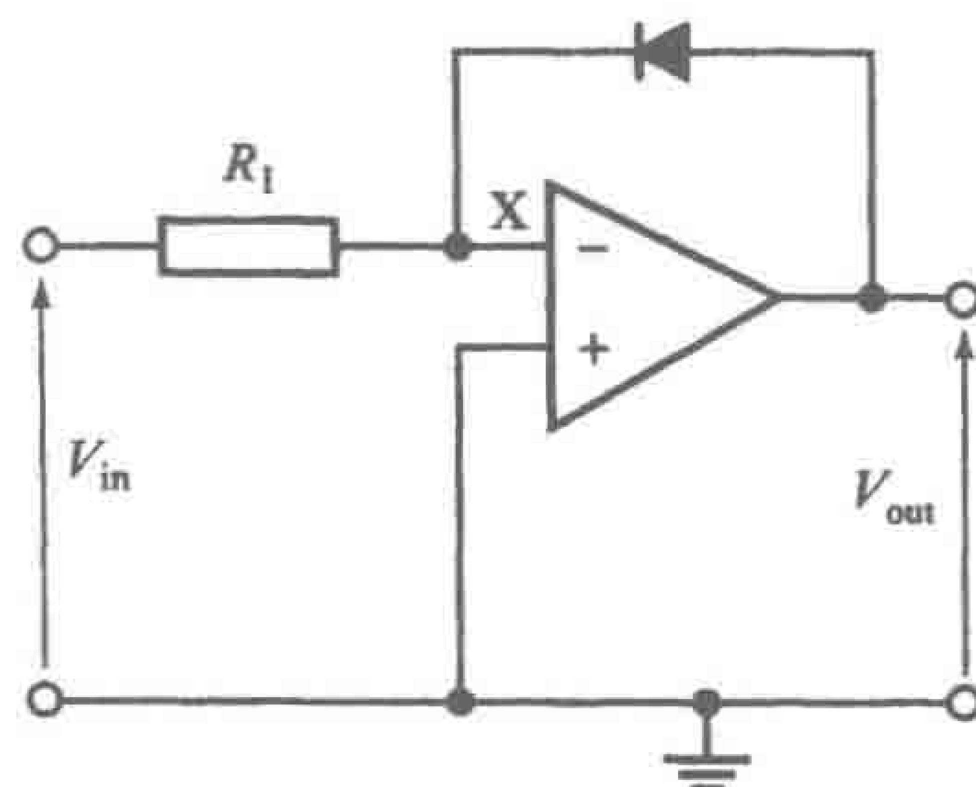


图 3-11　对数放大器

器所提供的，$V_{in} = Ae^{at}$，其中 A 和 a 都是常量，那么

$$V_{out} = K\ln V_{in} = K\ln(Ae^{at}) = K\ln A + Kat$$

结果就是 V_{out} 和 t 之间的线性关系。

78

3.2.7 比较器

比较器可以比较两个电压哪个大。一个没有反馈或其他元件的运算放大器就是比较器。将电压加载在同相输入端，另一个电压加载在反相输入端（见图 3-12a）。图 3-12b 展示了输出电压和两个输入电压差值之间的关系。当两个输入相等时没有输出。然而，当同相输入电压比反相输入电压稍微大一点时，输出跳变到一般为 +10V 的稳定正饱和电压。当反相输入比同相输入大时，输出就跳变到一般为 -10V 的稳定负饱和电压。这样的电路可以用来确定当一个电压超过某一电平值时，输出就会产生相应的反应。

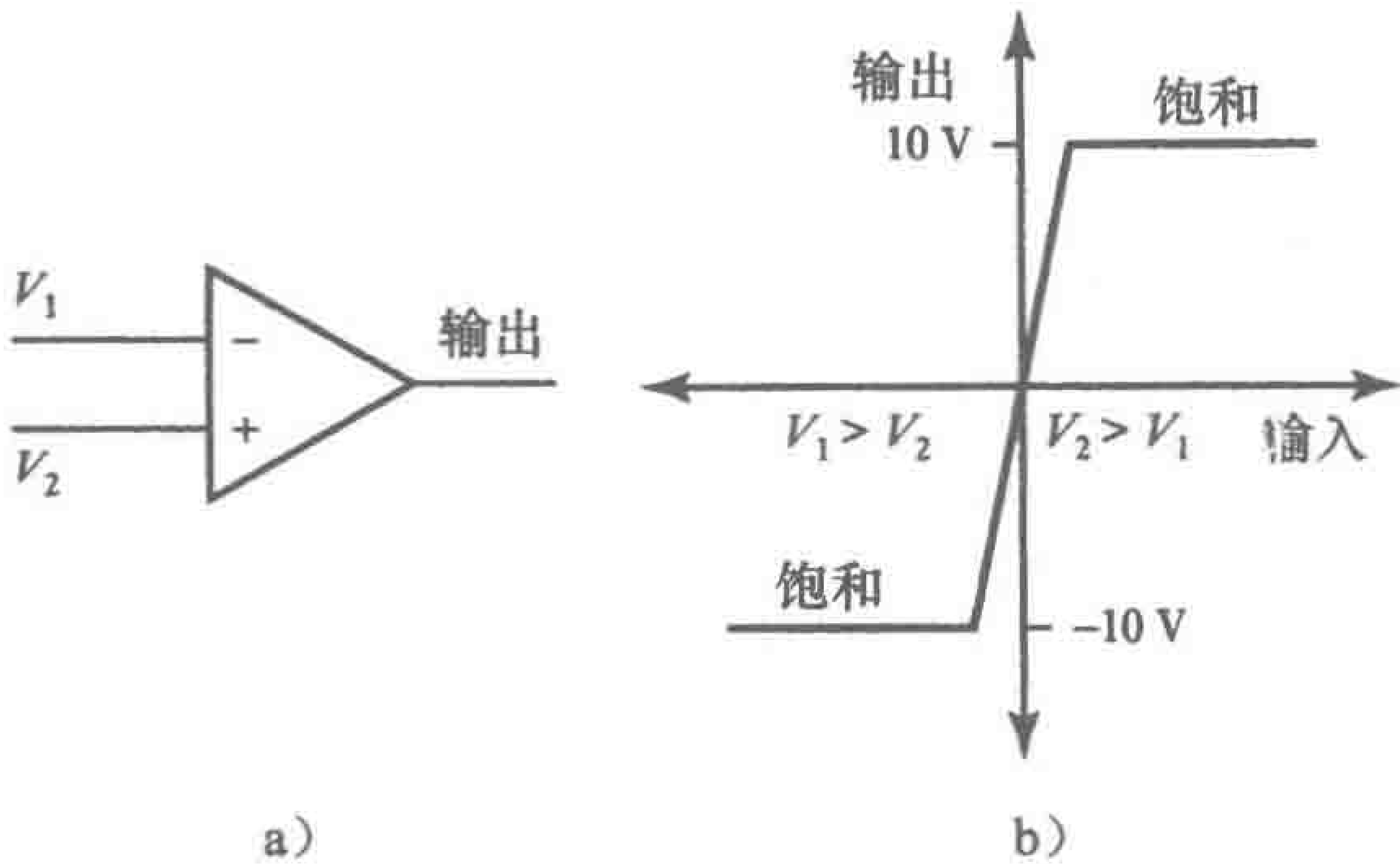


图 3-12 比较器

为了说明如何使用，考虑一个如图 3-13 所示的电路。它的设计意图是当达到临界温度时继电器闭合，然后来引发某些响应。这个电路中的惠斯通电桥的一条臂上有一个热敏电阻。电桥中的电阻阻值都是经过选定的，所以在临界温度时电桥平衡。当温度低于这一值时热敏电阻 R_1 比 R_2 高，电桥不平衡。结果在运算放大器的输入端之间将产生电势差，这会使得放大器的输出处于负饱和电压。这会使三极管处于关闭状态，即基极-发射极和基极-集电极反偏，没有电流流经继电器线圈。当温度升高时热敏电阻降低，电桥平衡，然后运算放大器转换到正饱和电压。这使得三极管开启处于正偏，线圈通电。

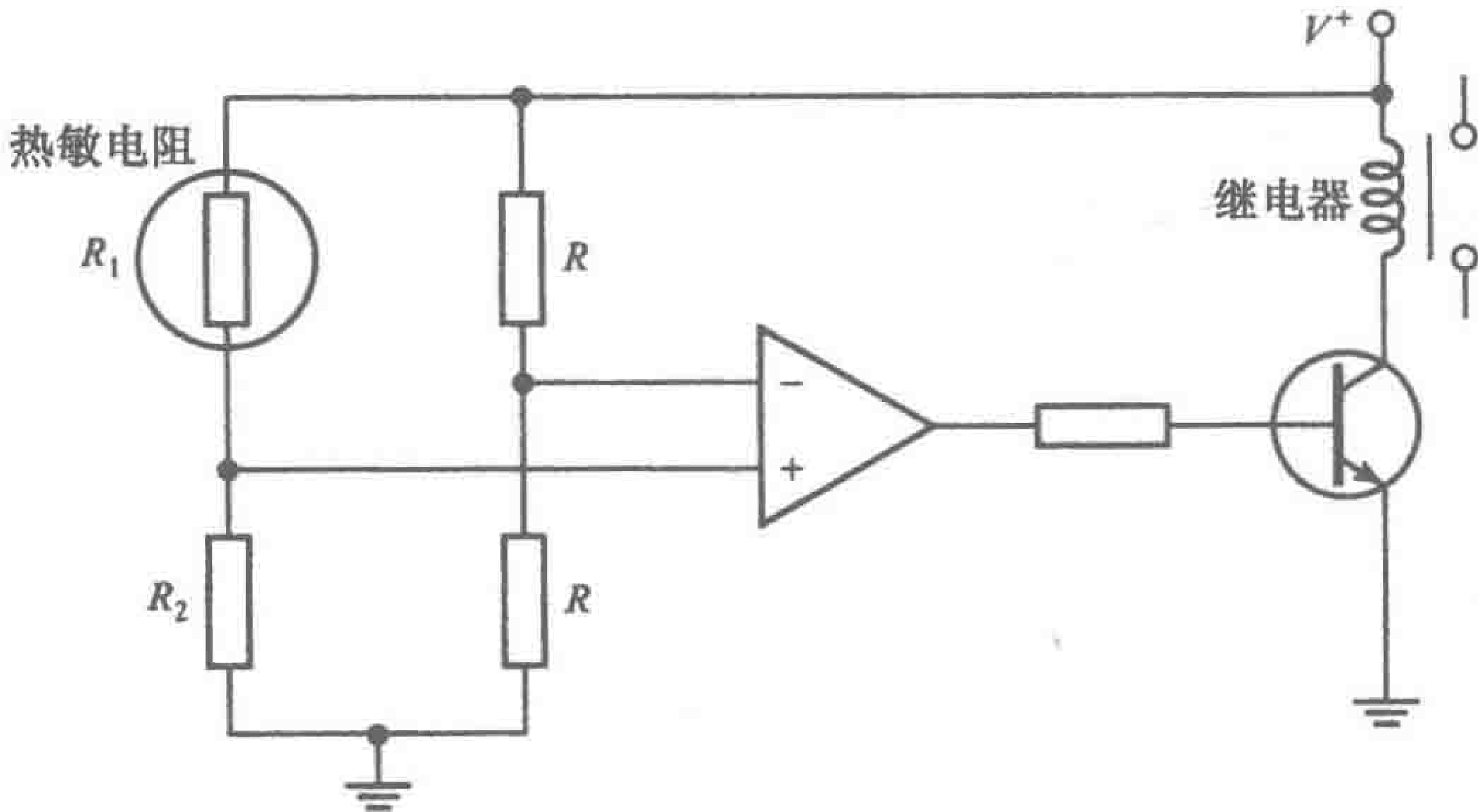


图 3-13 温度开关电路

作为比较器使用的另一个例子，考虑一个用于保证光盘播放机内的激光束能聚焦在光盘表面的系统。CD 机内有用于向 CD 聚焦激光束的透镜，声音信息存储在 CD 上一系列微观的凹坑

和平面所组成的序列中。从光盘上反射回来的光线将照在一个由四个光电二极管组成的阵列上（见图3-14）。这些光电二极管的输出将用来复原声音。使用四块光电二极管是因为阵列还可以用来判断激光束是否处于聚焦状态。当这些光束聚焦在光盘上时，会有一个圆形的光斑落在光电二极管阵列上，且光线均匀地分布在每个光电二极管上。这使得连接为比较器的运算放大器的输出为零。当光束未聚焦时，就产生一个椭圆光斑。这将导致每块光电管分到的光线不等。两组对角排列的光电管的输出是用来进行比较的，因为它们并不相等，所以比较器就给出了一个输出来指出光束未聚焦和哪一方向未聚焦。这一输出也可用于调整镜头，使光束在光盘上聚焦，来引导矫正行为。

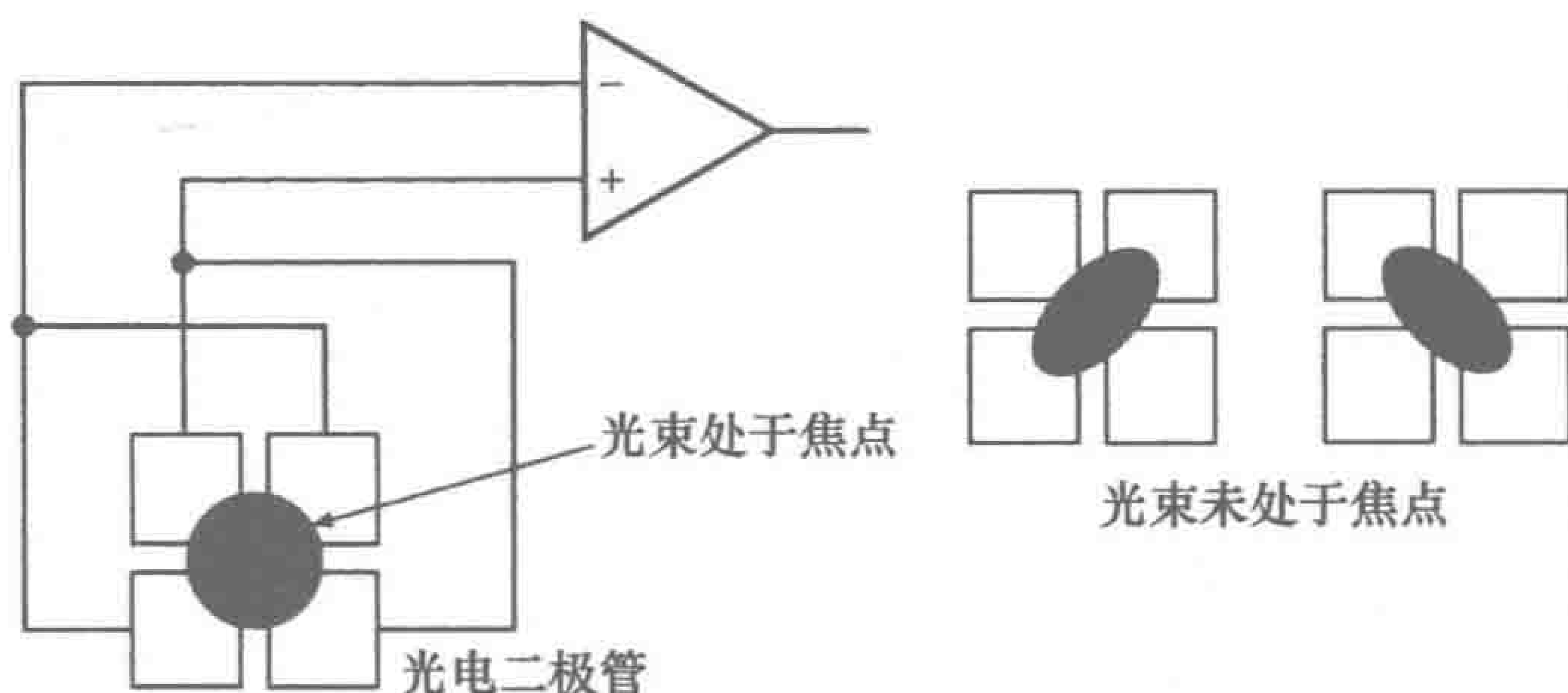


图3-14　CD播放器的聚焦系统

3.2.8　实际放大器

本章前几节在理想状态下讨论的运算放大器在现实中并不存在。一个特别重要的问题就是**偏移电压**（offset voltage）。

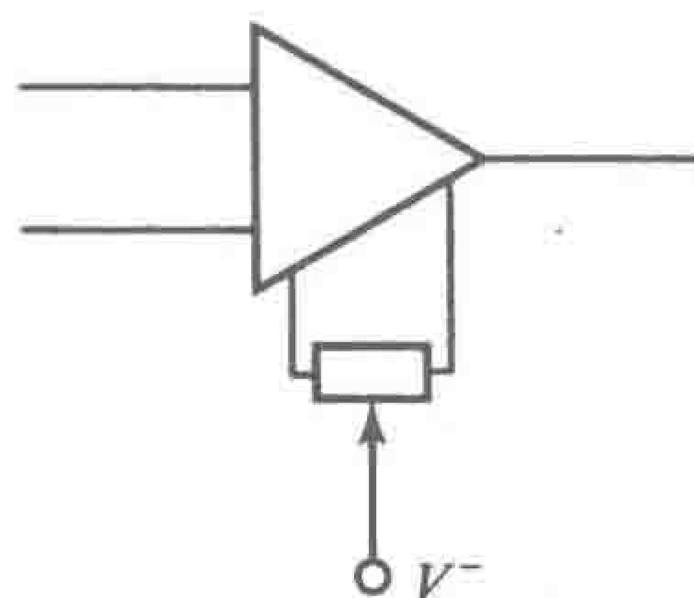


图3-15　偏移电压校正

运算放大器是放大两个输入之间差值的高增益放大器。因此如果两个输入短接了，我们会认为没有输出。然而在实际中这并不会发生，而且很可能会测出很大的输出电压。这是因为放大器中内部电路的不平衡所导致的。在输入端之间提供合适的电压可以使输出调零，这就是**偏移电压**。很多运算放大器都支持用来提供这一偏移电压的电位器。741通过在管脚1和5间（见图3-2）连接一个10kΩ的电位器并把电位器滑动端连接负电压源（见图3-15）来完成补偿。运算放大器内的不平衡可通过调节滑动端的位置，直到放大器达到无输入对应无输出的状态。一般通用放大器的偏移电压在1～5mV之间。

79～80

运算放大器为了偏置输入晶体管，需要在输入端引入少量的电流。偏置电流流经每个端口的源电阻时将会产生与输入串联的电压。理想状态下两个输入的偏置电流相等。然而实际上这并不会发生。因此这些偏置电流将会在没有输入信号、输出应为零时产生输出电压。当放大器由直流电压供电时会产生特殊的问题。两个偏置电流的均值称为**输入偏置电流**（input bias current），对于通用放大器，典型值为100nA。两个偏置电流的差值称为**输入偏移电流**（input-offset current）。理想情况下该值应该为零，但通用放大器都在10nA左右，约为输入偏置电流的10%～25%。

一个影响运算放大器在交流下使用的重要参数是**电压转换速率**（slew rate）。它是在标准阶跃响应下，电压随时间变化的最高转换速率。一般值在0.2V/μs到超过20V/μs不等。频率高时放大器的大信号运算就取决于输出能以多快的速率从一个电压转换成另一个。因此在有高频输入时，就需要有较高的电压转换速率。

举例来说，通用放大器LM348的开环电压增益为96dB，输入偏置电流为30nA，电压转换速率为0.6V/μs。宽频放大器AD711的开环增益为100dB，偏置电流为25pA，电压转换速率为20V/μs。

3.3 保护

在传感器与下一个单元的连接中存在着很多情况，比如一个微处理器有可能因为大电流或大电压而被烧坏的危险。为防止过高的电流，可以通过在输入端串联电阻和保险丝，电阻能把电流限制在可接受的范围，而在电流超过安全值时保险丝会断开。可以用齐纳二极管（zener diode）电路（见图3-16）来防止过高的电压和错误的极性。齐纳二极管工作时在达到击穿电压前和普通二极管基本一样。因此要使最大电压为5V，并且在电压超过5.1V时阻止电压传递，就要选用电压值为5.1V的齐纳二极管。当电压升高到5.1V时齐纳二极管击穿，它的阻值降低到很低的数值。结果二极管两端电压降低了，因此对下一个电路的输出也降低了。因为齐纳二极管是一种对一方向的电流有低电阻而对相反方向有高电阻的二极管，所以它也能用来防止错误极性的电压。当它和正确极性的电压连接时可以在输出端提供大电阻和高电压降。当供电极性反向，二极管为低阻值，因此输出端的电压也很低。

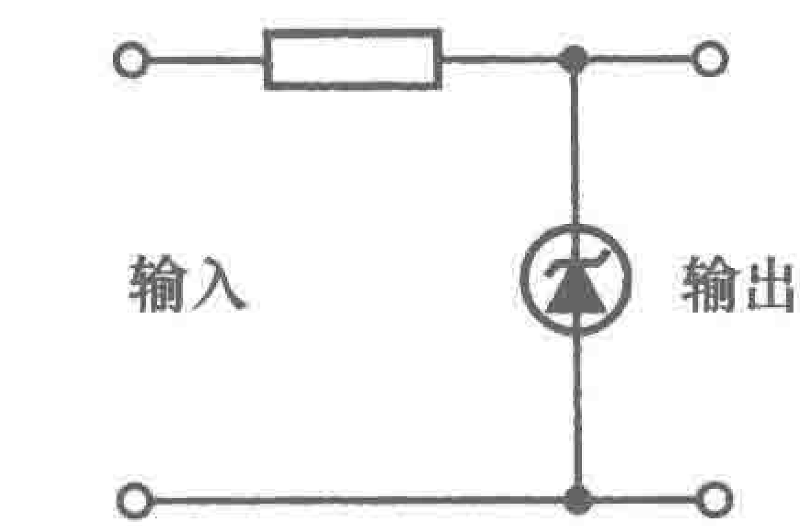


图3-16 齐纳二极管保护电路

81

在某些情况下，需要完全地隔离电路并移除它们之间的电气联系，这可以用**光隔离器**来完成。因此我们需要一个从微处理器到能发射红外辐射的发光二极管的输出。这些辐射可以通过光电晶体管或三端双向可控硅开关元件（triac）光隔离器来探测，并产生用来复现LED中电压变化的电流。图3-17展示了一些光隔离器的形式。传输率是用来描述输出电流和输入电流比率的术语。一般来说，一个简单的晶体管光隔离器（见图3-17a）的输出电流比输入电流小，转换率约为30%，最大值约为7mA。然而，达林顿形式（见图3-17b）可以提供比输入电流更大的输出电流，比如西门子6N139可提供800%的转换率，最大输出值为60mA。另一种光隔离器的形式（见图3-17c）利用了三端双向可控硅，因此可以用在交流电流上。一个典型的三端双向可控硅开关元件光隔离器可以在电源电压上使用。还有另外一种形式（见图3-17d）利用连接了零交叉单元的三端双向可控硅开关元件，比如摩托罗拉MOC3011，这可以减少瞬变和电磁干扰。

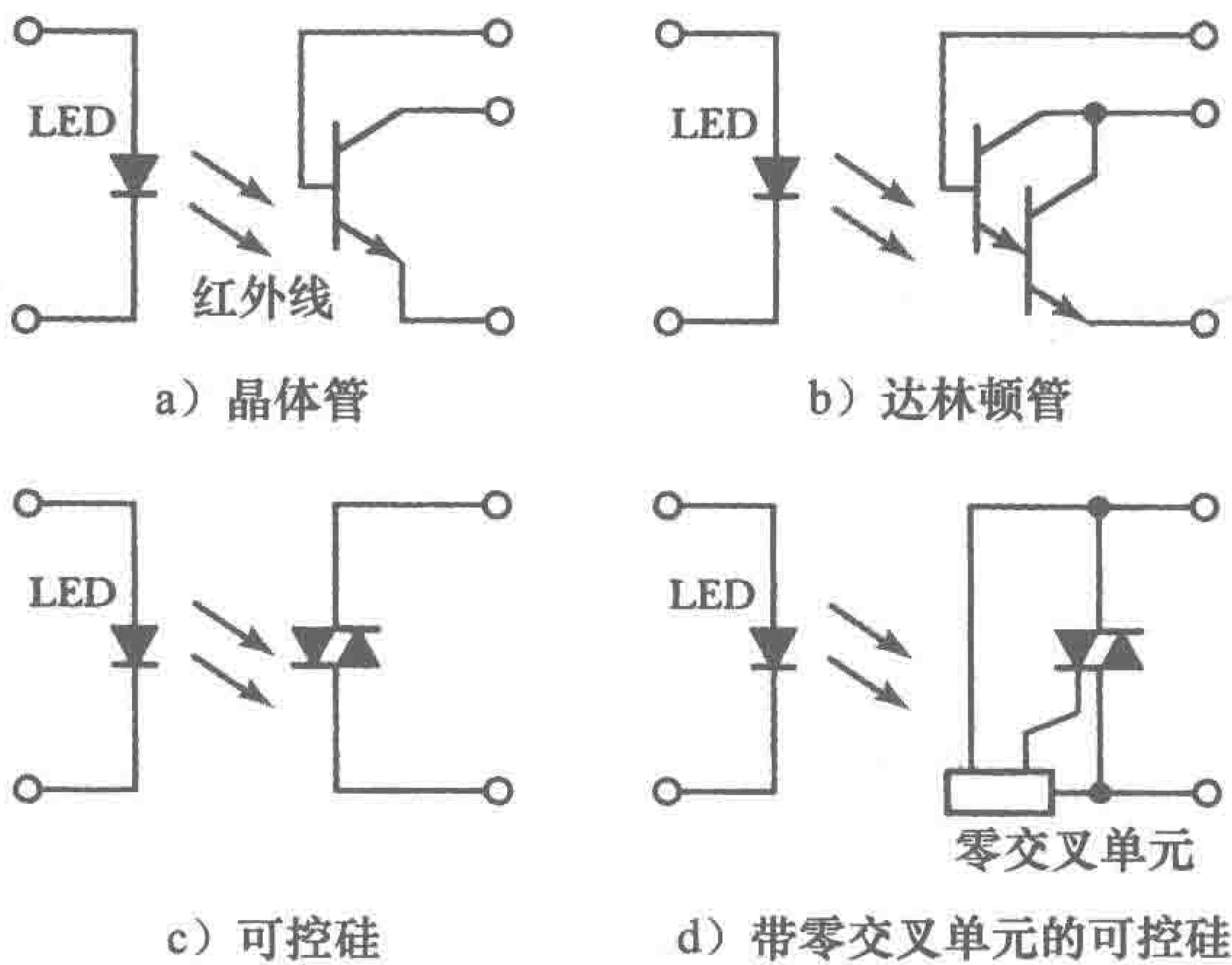


图3-17 光隔离器

光隔离器的输出可以直接用于切换低功率负载电路。因此达林顿光隔离器可以用来作为微处理器和灯泡或线圈间的接口。如果要接到大功率电路上，就需要光隔离器来控制继电

器，并利用继电器作为大功率设备的控制端。

微处理器输入的保护电路如图 3-18 所示。为了防止提供给 LED 的电压极性错误或电压过高，这里也用到图 3-16 所示的齐纳二极管电路来保护，如果输入为交变信号则会在输入端放置二极管来整流。

82

3.4 滤波

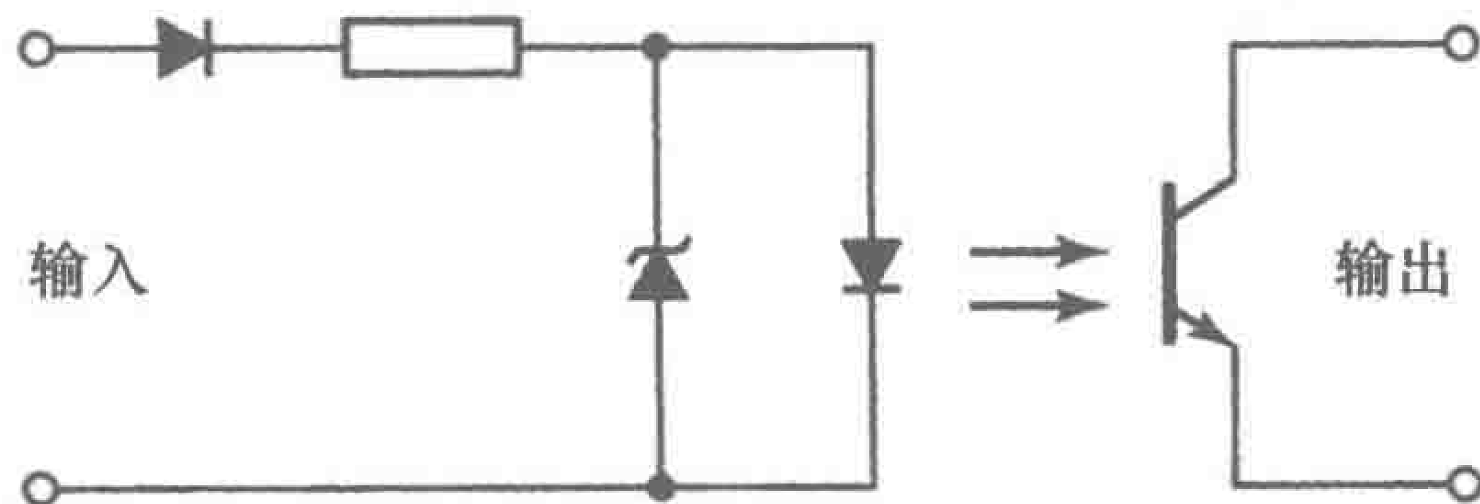


图 3-18 保护电路

滤波是一种滤除信号中某一频段，并允许其余频段通过的处理。能通过滤波器的频率范围称为**通带**（pass band），不能通过的称为**阻带**（stop band），通过和不通过之间的边界称为**截止频率**（cut-off frequency）。滤波器是用它们能通过或截止的频率范围来分类的。**低通滤波器**（见图 3-19a）的通频带能让从 0 到某一值的频率通过。**高通滤波器**（见图 3-19b）的通频带能让从某一值到无穷大的频率通过。**带通滤波器**（见图 3-19c）的通频带能让选定频域范围内的频率通过。**带阻滤波器**（见图 3-19d）的通频带能阻止特定频域范围内的频率通过。在所有情况下，截止频率都定义为当输出电压为通频带电压的 70.7% 时的频率。输入和输出功率的比值称为**衰减**（attenuation），它写为这一比值的对数，衰减的单位为贝尔（Bel）。因为这一单位很大，所以常用单位为分贝（decibels，dB），以分贝作为单位的衰减 = 10lg(输入功率/输出功率)。又因为电阻消耗的能量正比于电压的平方，所以以分贝作为单位的衰减 = 20lg(输入电压/输出电压)。输出电压为通频带电压的 70.7% 时衰减就为 3dB。

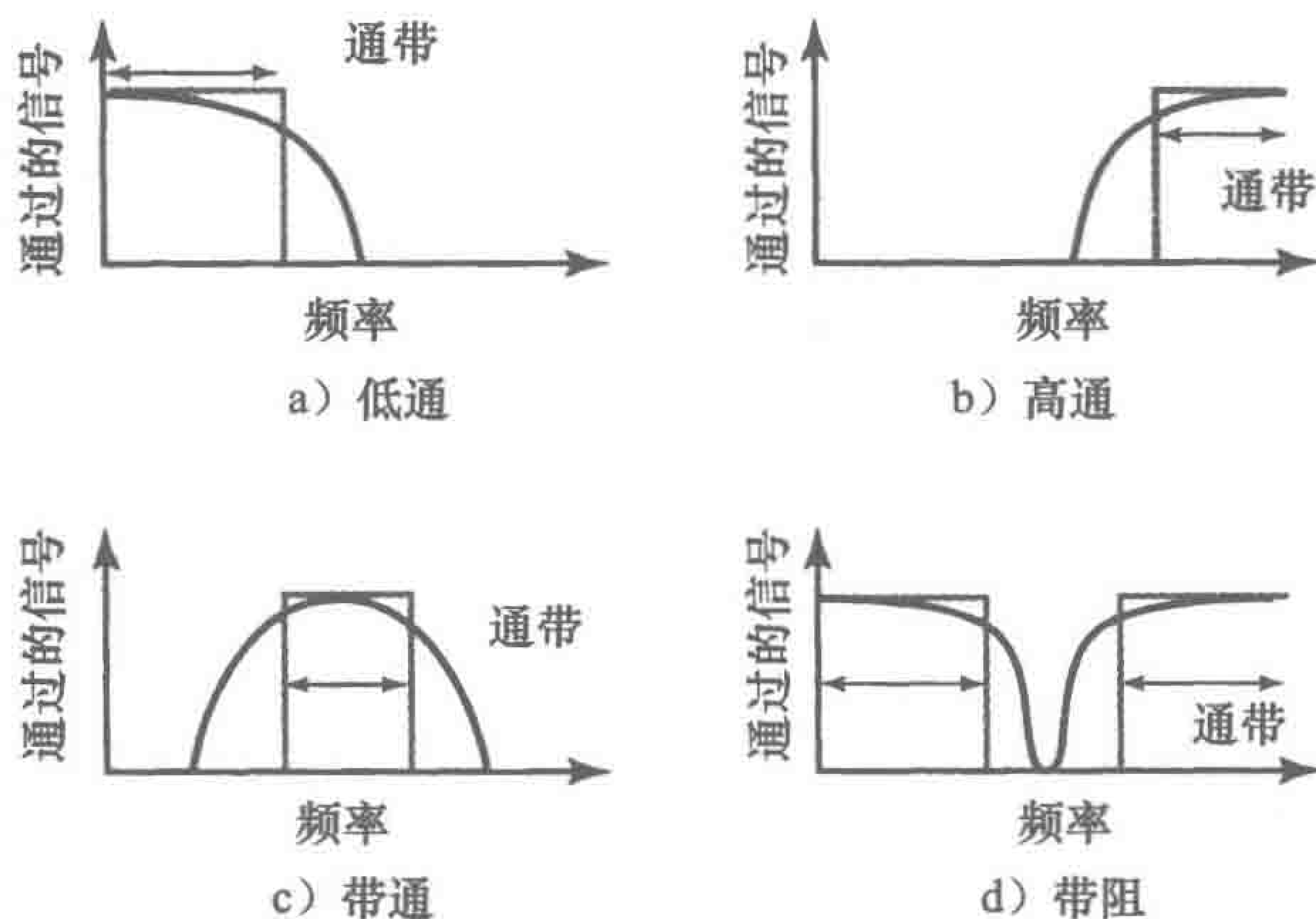
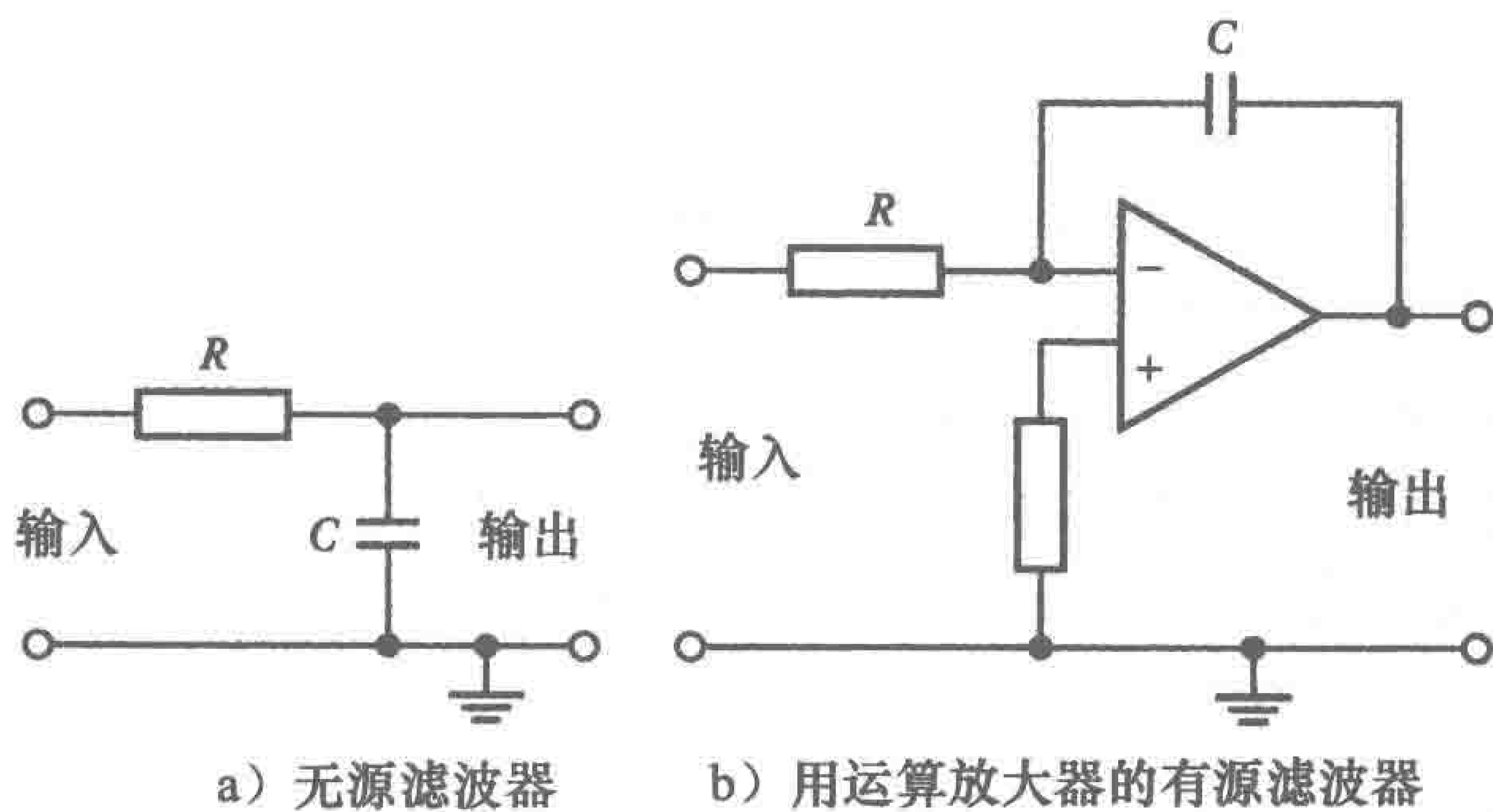


图 3-19 理想滤波器的特性

滤波器只由电阻、电容和电感组成的，称为**无源**（passive）滤波器，当滤波器包含运算放大器时称为**有源**（active）滤波器。无源滤波器的缺点是下一个单元的拉电流可能会改变滤波器的频率特性，而有源滤波器就没有这个问题。

低通滤波器是信号调理中常见的部件。这是因为大部分被传送的有用信息都是低频的。因为噪声一般都产生在高频，所以低通滤波器就可以用来阻断它们。因此一般可能会选用截止频率为 40Hz 的低通滤波器，可以阻挡交流电源的干扰信号和一般的噪声。图 3-20 展示了低通滤波器的基本使用形式。

83



a）无源滤波器 b）用运算放大器的有源滤波器

图3-20 低通滤波器

3.5 惠斯通电桥

惠斯通电桥可以用来把电阻变化转换为电压变化。图3-21展示了这一电桥的基本形式。当输出电压 V_o 为0时，B点的电势一定和D点相等。R_1 两端的电势差，即 V_{AB}，一定等于 R_3 两端的电势差，即 V_{AD}。因此 $I_1R_1 = I_2R_3$。这也表示 R_2 两端的电势差，即 V_{BC}，也一定等于 R_4 两端的电势差，即 V_{DC}。既然BD间没有电流流过，那么流经 R_2 的电流一定等于流经 R_1 的电流，而流经 R_4 的电流也一定等于流经 R_3 的电流。因此 $I_1R_2 = I_2R_4$。两式相除得

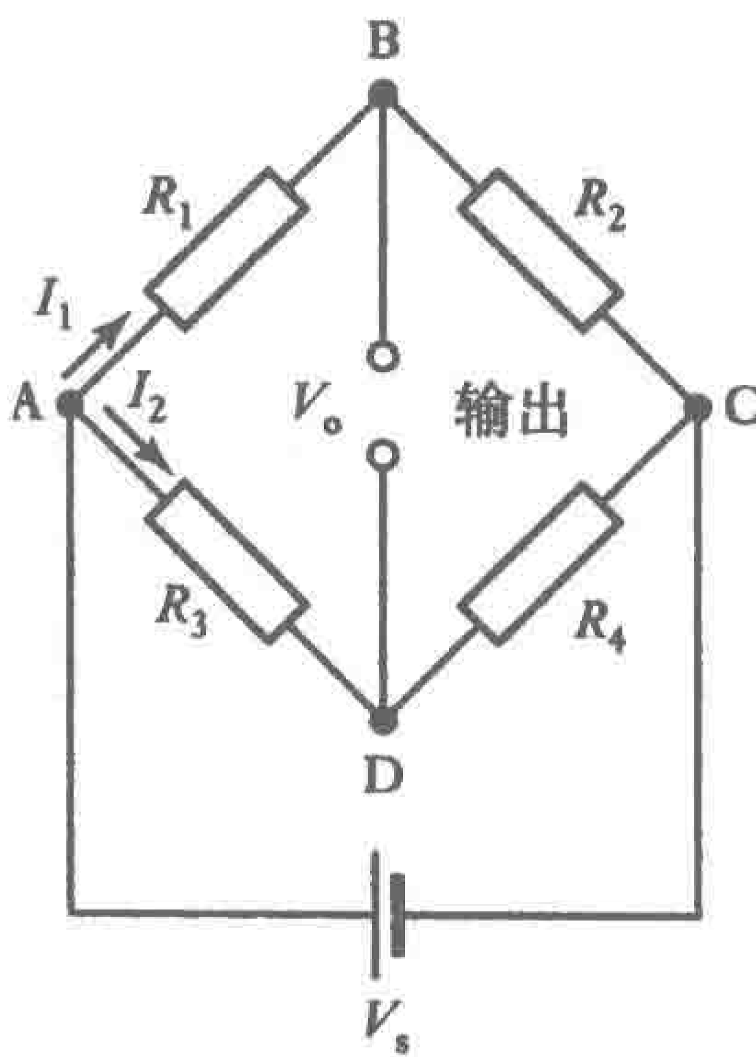


图3-21 惠斯通电桥

$$\frac{R_1}{R_2} = \frac{R_3}{R_4}$$

这样的电桥状态称为**平衡的**。

现在考虑当在平衡条件下改变其中一个电阻时会发生什么。电源电压 V_s 加在A点和C点间，因此 R_1 两端的电压为 $R_1/(R_1 + R_2)$ 乘以电源电压，即

$$V_{AB} = \frac{V_sR_1}{R_1 + R_2}$$

类似地，R_3 两端的电势差为

$$V_{AD} = \frac{V_sR_3}{R_3 + R_4}$$

因此B和D间的电势差，即输出电势差 V_o 为

$$V_o = V_{AB} - V_{AD} = V_s\left(\frac{R_1}{R_1 + R_2} - \frac{R_3}{R_3 + R_4}\right)$$

当 $V_o = 0$ 时，这个等式给出了平衡条件。

考虑电阻 R_1 为电阻可变的传感器。电阻从 R_1 变为 $R_1 + \delta R_1$ 会使输出从 V_o 变为 $V_o + \delta V_o$，其中

$$V_o + \delta V_o = V_s\left(\frac{R_1 + \delta R_1}{R_1 + \delta R_1 + R_2} - \frac{R_3}{R_3 + R_4}\right)$$

84

因此

$$(V_o + \delta V_o) - V_o = V_s\left(\frac{R_1 + \delta R_1}{R_1 + \delta R_1 + R_2} - \frac{R_1}{R_1 + R_2}\right)$$

如果 δR_1 比 R_1 小很多，那么上式就近似为

$$\delta V_o \approx V_S \left(\frac{\delta R_1}{R_1 + R_2} \right)$$

通过这个近似式可知，输出电压的变化正比于传感器电阻的变化。这是在输出没有负载电阻时的输出电压。如果有负载电阻时，那么就要考虑到负载效应。

为了说明上面所说的，考虑一个铂电阻温度传感器，它的电阻在0℃时为100Ω，并构成惠斯通电桥的一条臂。电桥的其他臂的阻值也为100Ω，在这一温度下电桥平衡。如果铂电阻的温度系数为0.0039/K，并假设输出的负载可以为无限大，那么温度升高1℃电桥的输出电压会怎样变化呢？设忽略内部电阻时的电源电压为6V。于是，铂电阻随温度的变化可写为：

$$R_t = R_0(1 + \alpha t)$$

R_t 为 t（℃）时的电阻，R_0 为0℃时的电阻，α 为温度系数。因此

$$\text{电阻变化} = R_t - R_0 = R_0 \alpha t$$

$$= 100 \times 0.0039 \times 1 = 0.39\Omega/\text{K}$$

既然电阻变化相对于100Ω很小，那么近似式就可以使用。因此

$$\delta V_o \approx V_S \left(\frac{\delta R_1}{R_1 + R_2} \right) = \frac{6.0 \times 0.39}{100 + 100} = 0.012\text{V}$$

3.5.1 温度补偿

在很多包含电阻式传感器的测量中，实际传感元件可能会处于长导线的末端。除了传感器外，导线的电阻也会受到温度变化的影响。比如说，一个铂电阻温度传感器在导线末端包含一个铂线圈。当温度改变时，不只是线圈的电阻会变化，导线电阻也会改变。我们只需要线圈的变化，所以就需要一些补偿导线电阻的措施。一种方式就是用三条导线连接在线圈上，如图3-22所示。线圈这样连接到惠斯通电桥上：导线1串联电阻 R_3，而导线3串联铂电阻线圈 R_1，导线2连接电源。因为它们用的是金属、直径和长度都一样且三条导线都靠在一起，所以导线电阻的任何变化都会同等地影响全部三条导线。这使得导线电阻变化在电桥两臂中相同，并当 R_1 和 R_3 相等时它们就互相抵消。

85

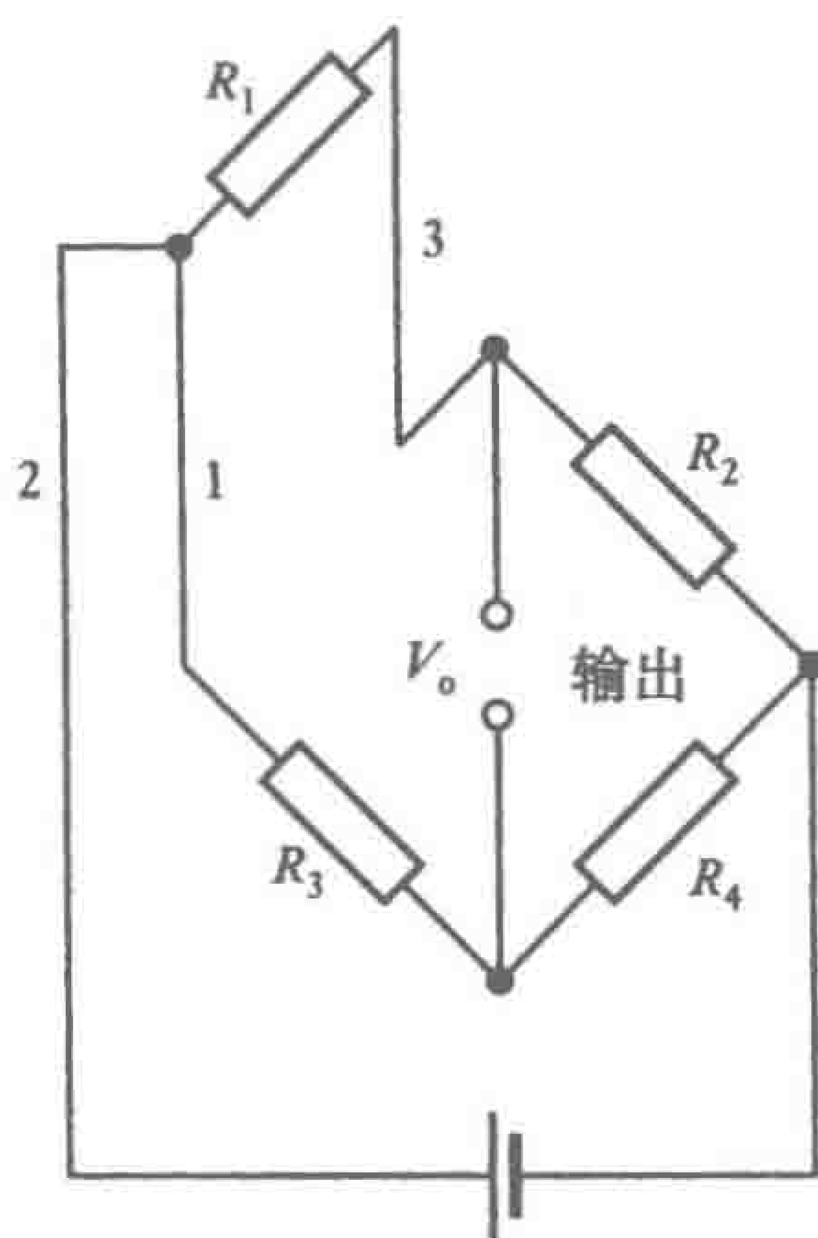


图3-22 端点补偿

电阻应变片是另外一种需要对温度影响进行补偿的传感器。当所受张力改变时，应变片的电阻就改变。然而在温度变化时它也会改变。一种消除温度影响的方法是使用**补偿应变片**（dummy strain gauge）。这是一块和受到张力的工作应变片相同的应变片，安装在同样的材料上但不受张力影响。它放置在工作的应变片边上，所以温度变化也与其相同。因此温度的变化会使得两块应变片改变同样的电阻。工作应变片安装在惠斯通电桥的一条臂上（见图3-23a），而补偿应变片安装在另一条臂上，这样由温度引起的电阻变化的影响就消除了。

应变片还常和称重传感器或膜片压力计等传感器一起使用来测量位移量。这种情况也需要使用温度补偿。当可以使用补偿应变片时，最好的方法就是使用4块应变片。它们的粘贴方式使其中两块受的是张力，而另外两块受的是压力。图3-23b中的称重传感器就展示了这种安装方法。应变片在受拉时会增大电阻而受压时减小电阻。当这些应变片被连接在惠斯通电桥的4条臂中时（见图3-23b），由于它们受到温度变化的影响是相同的，所以装置达到了温度补偿。这也使输出电压比单个应变片时大很多。

为了说明上述内容，考虑一个如图3-23排列的4个应变片的称重传感器，由4个工作应变片组成了电桥。应变片灵敏度为2.1，阻值为100Ω。当称重传感器加载压力时，竖直的应变片

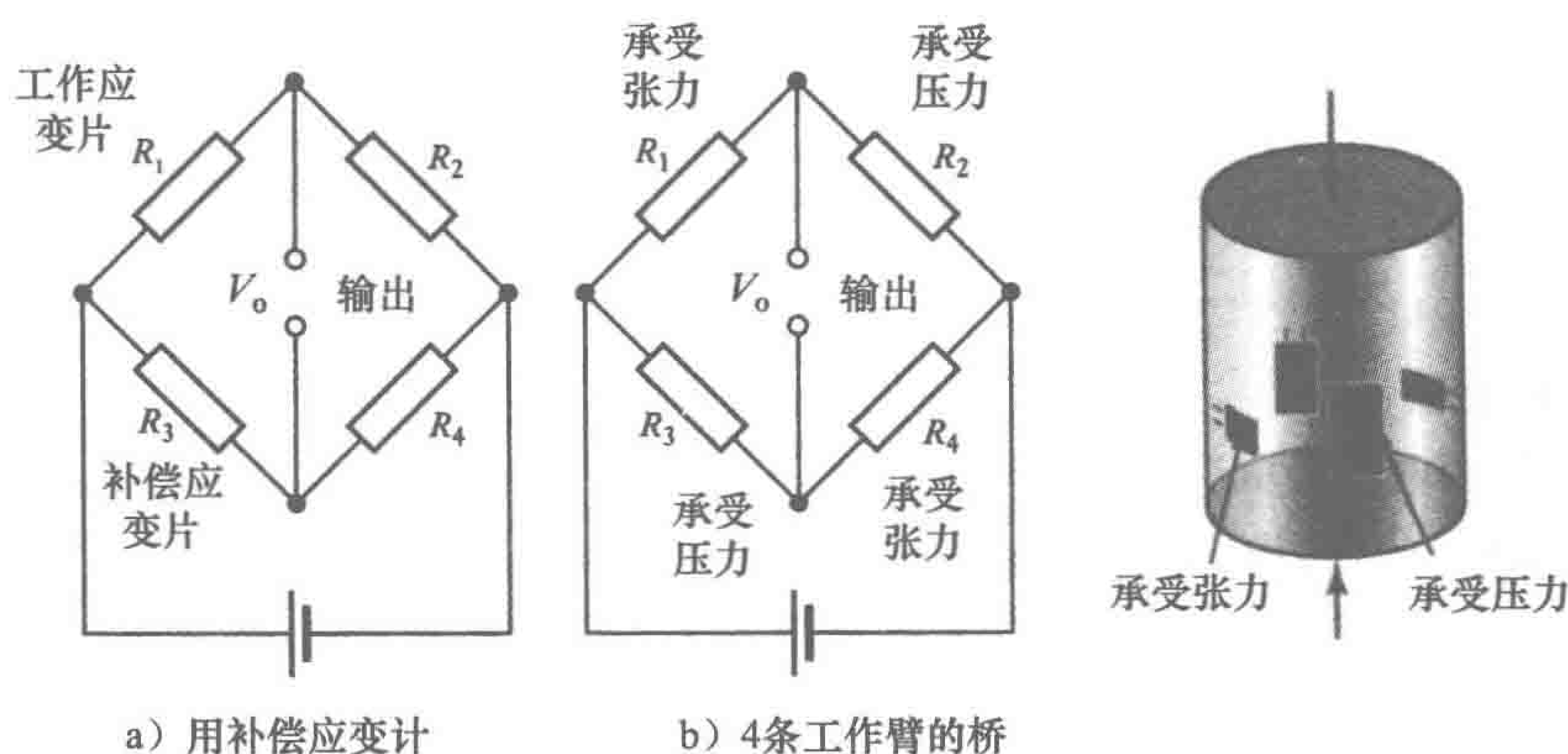


图3-23 应变片的补偿

受压，因为当物体受压时就会产生侧向伸展，所以水平的应变片就会受到张力（横纵受力的比值称为泊松比，一般为0.3左右）。因此如果压缩应变片受到 -1.0×10^{-5} 的力时，张力应变片就受到 $+0.3\times10^{-5}$ 的力。电源电压为6V，电桥的输出电压被差分运算放大器电路放大，如果想要为负载提供1mV的输出，那么反馈电阻和输入电阻的比值该为多少？ 86

应变片电阻变化与压缩应变的关系为 $\Delta R/R=G\varepsilon$：

$$\text{电阻变化}=G\varepsilon R=-2.1\times1.0\times10^{-5}\times100$$
$$=-2.1\times10^{-3}\Omega$$

对于张力应变片有

$$\text{电阻值变化}=G\varepsilon R=2.1\times0.3\times10^{-5}\times100$$
$$=6.3\times10^{-4}\Omega$$

不平衡电势差（见3.5节的前面部分）就为

$$V_o=V_s\left(\frac{R_1}{R_1+R_2}-\frac{R_3}{R_3+R_4}\right)$$
$$=V_s\left(\frac{R_1(R_3+R_4)-R_3(R_1+R_2)}{(R_1+R_2)(R_3+R_4)}\right)$$
$$=V_s\left(\frac{R_1R_4-R_2R_3}{(R_1+R_2)(R_3+R_4)}\right)$$

现在我们假设每个电阻发生变化。然而，因为电阻的变化相对于两个电阻的和实在太小了，所以我们可以忽略分母中的变化。因此有：

$$V_o=V_s\left(\frac{(R_1+\delta R_1)(R_4+\delta R_4)-(R_2+\delta R_2)(R_3+\delta R_3)}{(R_1+R_2)(R_3+R_4)}\right)$$

忽略 δ 量的乘积产生的微小量，又由之前我们得到的电桥平衡条件 $R_1R_4=R_2R_3$，得

$$V_o=\frac{V_sR_1R_4}{(R_1+R_2)(R_3+R_4)}\left(\frac{\delta R_1}{R_1}-\frac{\delta R_2}{R_2}-\frac{\delta R_3}{R_3}+\frac{\delta R_4}{R_4}\right)$$

因此

$$V_o=\frac{6\times100\times100}{200\times200}\left(\frac{2\times6.3\times10^{-4}+2\times2.1\times10^{-3}}{100}\right)$$

所以输出为 8.19×10^{-5}V。这成为差分放大器的输入。因此，利用3.2.5节推出的等式，

$$V_o = \frac{R_2}{R_1}(V_2 - V_1)$$

$$1.0 \times 10^{-3} = \frac{R_2}{R_1} \times 8.19 \times 10^{-5}$$

87 所以 $R_2/R_1 = 12.21$ 。

3.5.2 热电偶补偿

热电偶产生的电动势取决于它两端（详见2.9.5节）的温度。理想情况下，如果冷端保持在0℃，那么温度与电动势的关系就能直接地从表格得出。然而，这并不总是可行的，冷端通常是处于室温下。这就需要在热电偶上加载电势差来补偿。这一电势差必须和热电偶一端处于0℃而另一端处于室温时产生的电势差相等。这一电势差可以通过在惠斯通电桥中使用热电阻传感器来获得。电桥在0℃下为平衡态，故电桥的输出电压提供了在另一温度下的修正电势差。

金属电阻温度传感器的电阻可表示为

$$R_t = R_0(1 + \alpha t)$$

其中 R_t 为 t℃时的电阻，R_0 为0℃时的电阻，α 为电阻的温度系数。因此

$$\text{电阻变化} = R_t - R_0 = R_0\alpha t$$

把 R_1 作为电阻温度传感器的电桥输出电压为

$$\delta V_o \approx V_S\left(\frac{\delta R_1}{R_1 + R_2}\right) = \frac{V_S R_0 \alpha t}{R_0 + R_2}$$

当温度在小范围内变化时，比如从0℃到室温，热电势 e 随温度变化关系可以认为是线性的。因此 $e = kt$，k 为常量，即温度每变化1℃时产生的电势差。所以，为了补偿，我们有：

$$kt = \frac{V_S R_0 \alpha t}{R_0 + R_2}$$

故

$$kR_2 = R_0(V_S\alpha - k)$$

对于铁－康铜热电偶 k 为 51μV/℃，补偿可由一个镍电阻来提供，其电阻在0℃时为10Ω，电阻的温度系数 α 为0.0067/K，提供给电桥的电源电压为1.0V，则可得到 R_2 为1304Ω。

3.6 脉冲调制

在转换从传感器传来的小幅度直流信号时常常会遇到这样的问题，用来放大它们的运算放大器的增益可能漂移，从而导致输出也产生漂移。如果信号是脉冲序列而不是连续信号，就能解决这一问题。

88 实现这样转换的一种办法就是利用图3-24所建议的方法对直流信号进行斩波处理。从斩波器发出的输出是一系列脉冲，它们的高度和输入信号的直流电平有关。这一步骤称为**脉冲振幅调制**。在经过放大或其他信号调理后，调制信号可以被解调并作为直流输出。脉冲振幅调制所得的脉冲高度和直流电压的大小有关。

脉冲宽度调制（Pulse Width Modulation，PWM）在控制系统里被广泛地用来控制直流电压的均值。如果有一个被斩波成脉冲的常数值模拟电压，则通过改变脉冲宽度就能改变其电压的均值（如图3-25所示）。每一周期内高电平持续的时间与总周期的比值称为**占空比**（duty cycle）。因此对于PWM信号，高电平信号在每一周期内为一半时，占空比为1/2或50%。如果每一周期只占1/4，则占空比为1/4或25%。

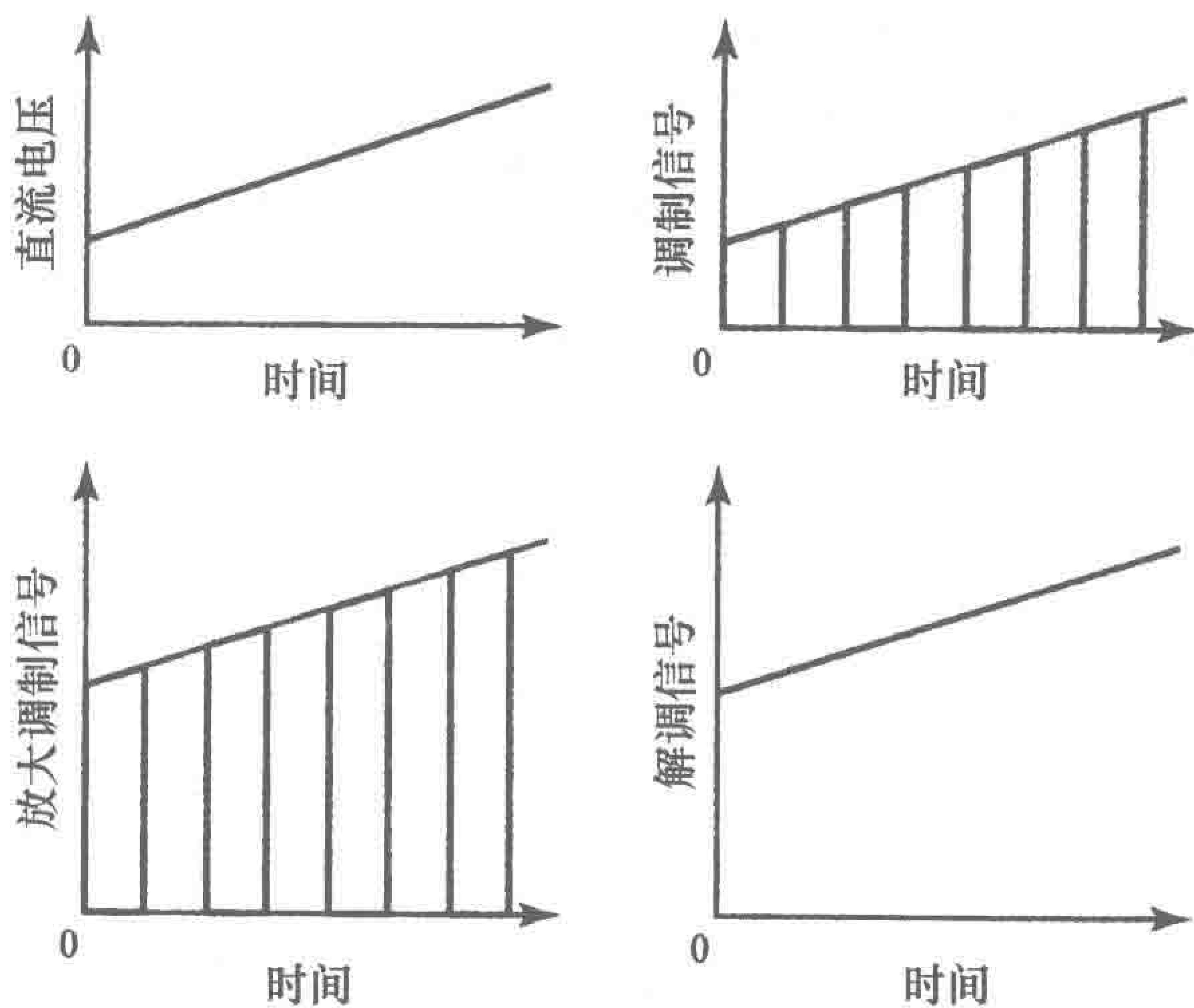


图3-24 脉冲振幅调制

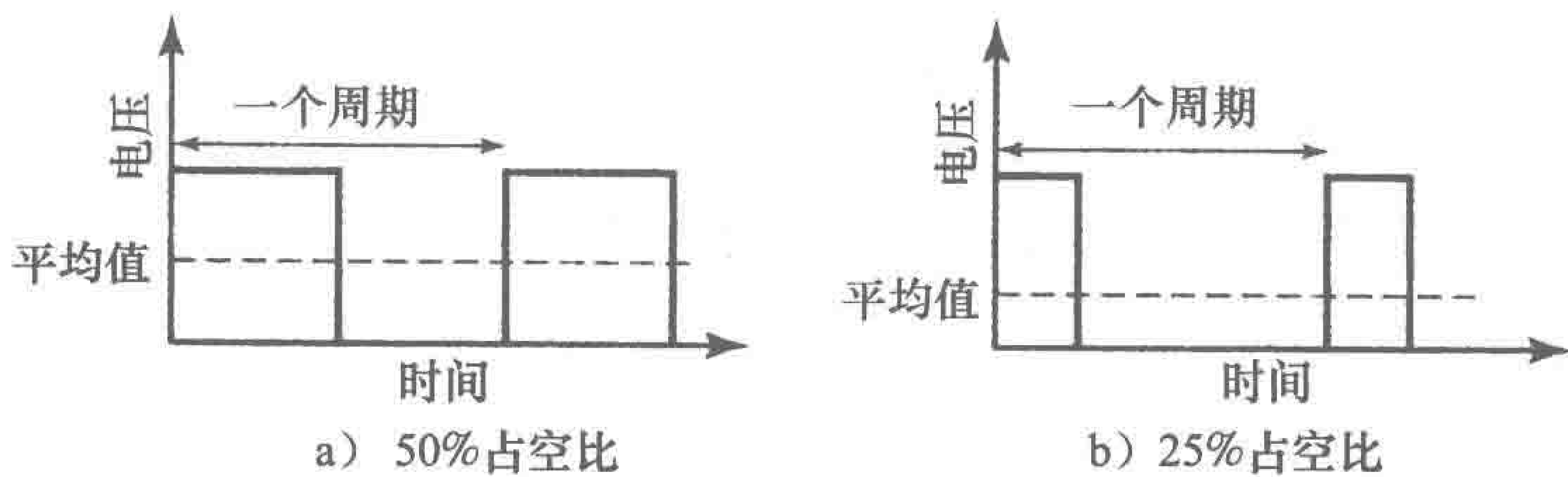


图3-25 电压控制的PWM

3.7 与信号相关的问题

当把传感器连接到信号调理设备或控制器上时，由于接地和电磁干扰的原因，信号可能会产生一些问题。

3.7.1 接地

89

一般从传感器和信号处理设备来的信号都是以电压形式传送到控制器上。这些电压都是两点间的电势差。如果其中一点接地则称为**接地信号源**。如果都没有接地则称为**浮动信号源**。接地源的电压输出是地和信号源正极信号线之间的电势差。如果是浮动信号源，信号源就没有任何绝对值作参考，每条电压线都可能与地有电势关系。

差分系统，比如一个差分放大器，考虑的是两条输入导线之间存在的电势差。如果都有相对地的参考电压，V_A 和 V_B，则**共模电压**为它们的均值，即 $\frac{1}{2}(V_A + V_B)$。因此，如果一条输入线为10V、另一条为12V，则电势差为2V，共模电压为11V。差分测量系统与两个输入的差值 $V_A - V_B$ 有关而与共模电压无关。然而共模电压会影响得到的电势差值，它对差值的作用程度通过共模抑制比（CMRR）来描述（详见3.2.5节）。它是系统差分增益和共模增益的比值，当用分贝表示时，可写为20lg(差分增益/共模增益)。CMRR越高，则相对于共模电压增益，差分增益就越大，共模电压的作用就越小。当一个差分放大器的CMRR为10 000或80dB时，如果需要的差分信号和共模电压的值一样大，则其输出的电压就会是共模电压输出的10 000倍。

当电路存在着多个接地点时，系统也会产生问题。比如，传感器和信号调理器可能同时接

地了。在大型的系统中，重复接地是难以避免的。然而，在两个接地点之间可能会有电势差，因此就会有明显的电流流过接地点间的微小却有限的地电阻（见图3-26）。这称为**接地环路电流**（ground-loop current）。两个接地点间的电势差可能不仅仅是直流的，也有可能是交流的，比如交流电源的交流噪声。还有一个问题就是，这一环路中的电流会与临近的电路产生电磁耦合。因此由于接地环路的存在会使远程测量变得困难。

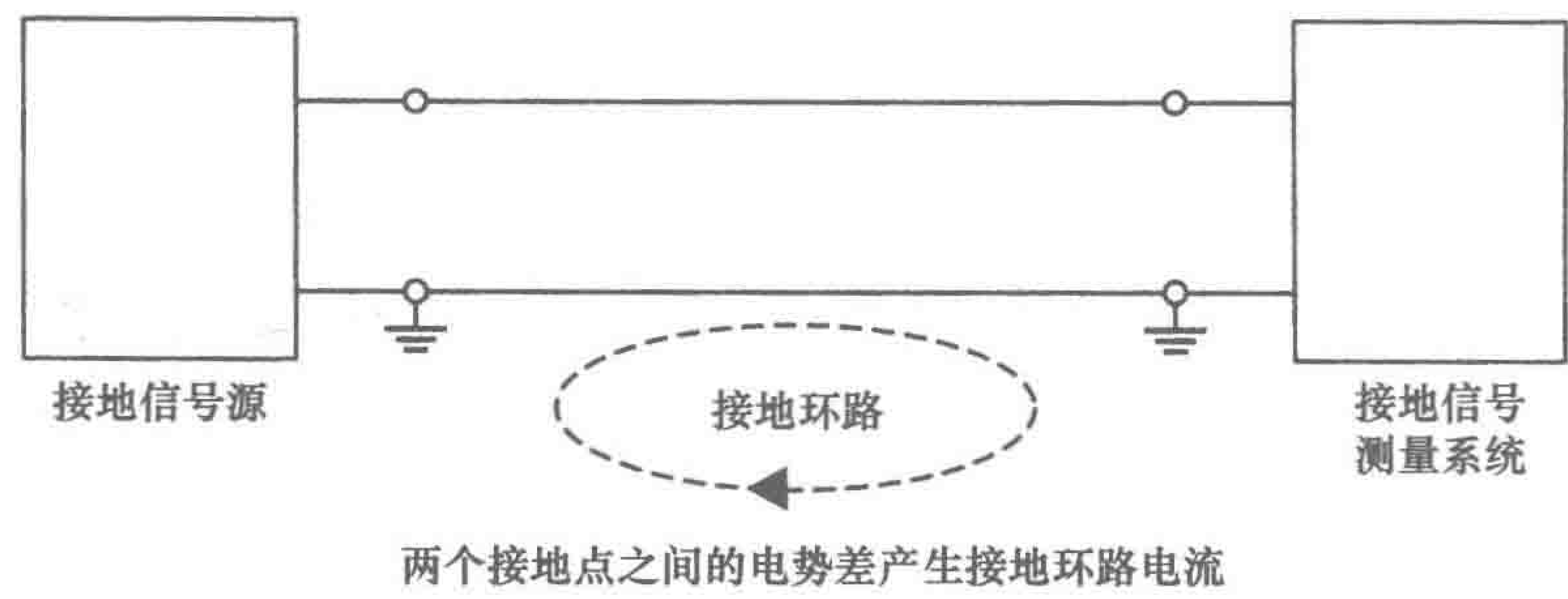


图3-26　接地环路

90

如果多个接地点靠在一起，并且地电阻小到能使得接地点间的电压降可被忽略，则多点接地产生的接地环路就能够减到最小。如果信号源和测量系统之间存在电气隔离，那么就能消除接地环路。这可以通过光隔离器（详见3.3节）或变压器（见图3-27）来实现。

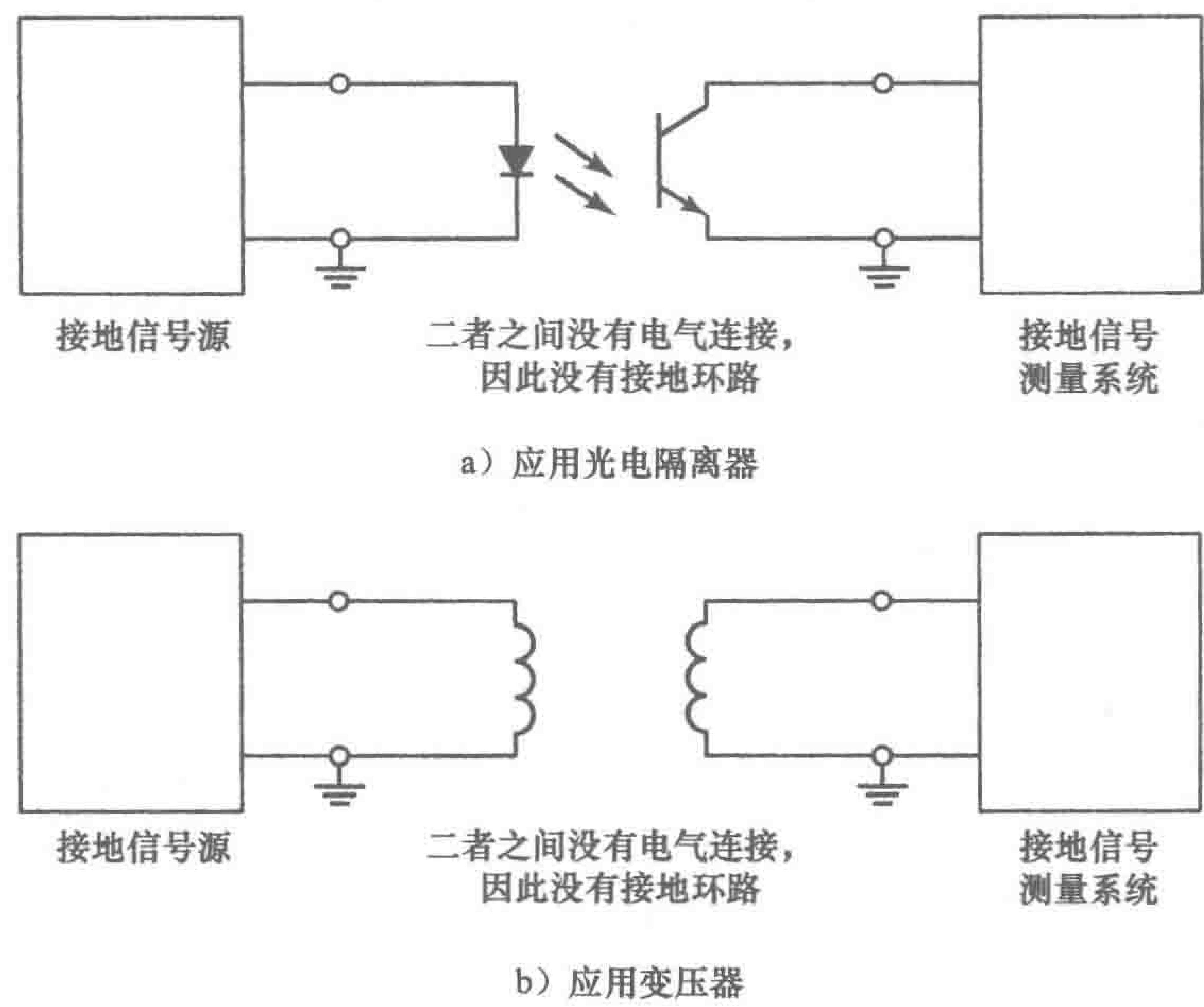


图3-27　隔离

3.7.2　电磁干扰

电磁干扰是电路中由于时变的电磁场产生的不良影响。这些干扰源一般为日光灯管、直流电动机、继电器线圈、家用电器和汽车的电气系统。

静电干扰是由相邻的导体之间产生的电容所引起的。静电屏蔽可以防止它的发生。它是由导电金属（比如铜或铝）制成的屏障，用来包裹住导体或电路。因此可能会在传感器和测量系统之间通过屏蔽电缆来连接。如果传感器接地，那么屏蔽就需要在传感器接地的同一点与之

连接，这样就能最大限度地减小接地环路（见图3-28）。

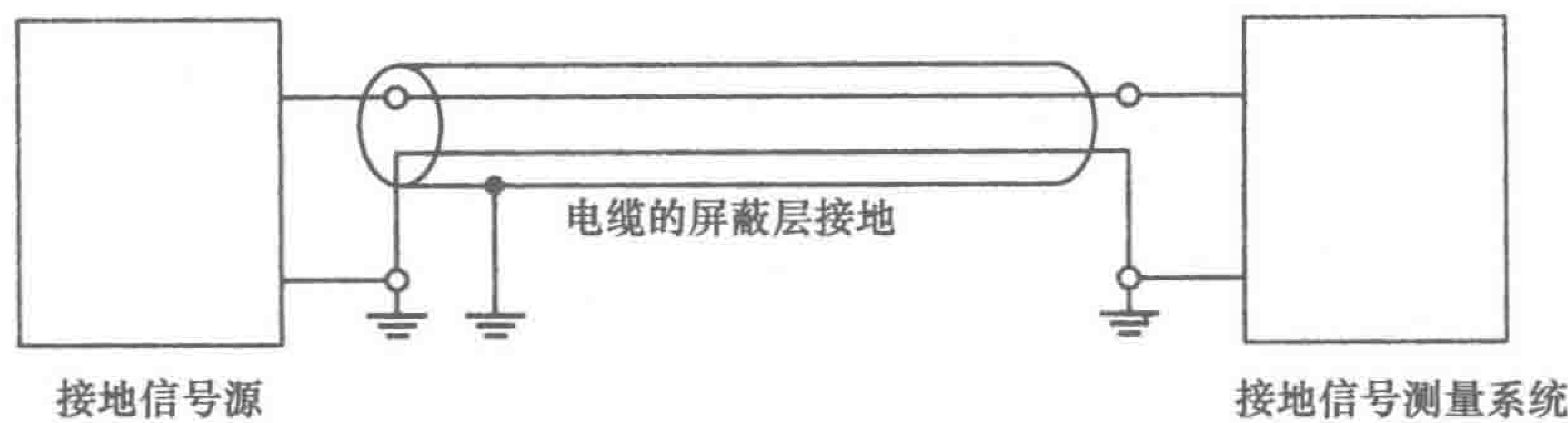


图3-28 使用屏蔽电缆来减少静电干扰

91

变化的磁场也会产生干扰，它会在测量系统中感应出电压。对此可以通过尽可能将元件远离干扰源安置、尽量减小系统中环路的面积和使用双绞线来相互连接（见图3-29）等方法来保护。在双绞线中，相邻双绞线在相位上耦合交替，从而消除了它们产生的影响。



图3-29 用双绞线来减少电磁干扰

3.8 功率输送

在控制系统中部件的相互连接存在很多不同的情况。例如，对于电器元件我们会把一个传感器系统连接到放大器；对于机械系统我们会用电动机来转动负载。而两个元件之间达到最大能量传输的条件是我们所关心的。

作为介绍，考虑一个电动势为 E，内阻为 r 的直流电源，并有一个电阻为 R 的负载（见图3-30）。提供给负载的电流为 $I = E/(R+r)$，故提供给负载的功率为

$$P = I^2R = \frac{E^2R}{(R+r)^2}$$

当 $dP/dR = 0$ 时给负载提供的功率达到最大。

$$\frac{dP}{dR} = \frac{(R+r)^2E^2 - E^2R2(R+r)}{(R+r)^4} = E^2\frac{[(R+r)-2R]}{(R+r)^3}$$

图3-30 提供负载的直流源

当 $(R+r) = 2R$ 时该式为零，故最大功率传输的条件就是 $R = r$，即电源和负载的电阻相等时。

当一个有着内阻的交流电源为一个负载电阻供电时，相似地可以得到最大功率传输的条件是电源和负载阻抗相等。比如电子系统中的高阻抗传感器，在电源和负载之间就需要使用一个阻抗与其相匹配的放大器来使功率传输最大化。这样的放大器一般为高输入阻抗、低输出阻抗的高增益放大器。

本章小结

信号调理包含了为防止系统中元件损坏而施加的保护措施、把信号转换为需要的形式、调节信号到合适的电平、减小噪声、线性化信号等。

92

常用的信号调理元件是**运算放大器**，它是一种能有100 000倍或更高数量级的高增益直流

放大器。

通过加入电阻和保险丝作为**保护**能防止高电压或高电流；齐纳二极管可以用来防止电压极性错误和过高电压。光隔离器可以用来完全隔离电路，切断它们之间的全部电气连接。

滤波器用来去除信号中的特殊频带，并使其余频带能够传输。

惠斯通电桥用来把电阻值的变化转换成电压的变化。

当把传感器连接到信号调理设备和控制器时，当电路存在多个**接地**点和时变电磁场引发的**电磁干扰**会产生很多问题。

最大化电气元件间的**能量传输**需要使阻抗匹配。

习题

3.1　设计一个运算放大器电路，当输入从0增加到100mV时电路产生0～-5V的输出。

3.2　一个反相放大器的输入电阻为2kΩ。要使电压增益为100，试确定需要的反馈电阻。

3.3　设计一个加法放大器电路，当输入从0增加到100mV时电路产生的输出为-1～-5V。

3.4　如图3-8所示，一个差分放大器与一个热电偶传感器一起使用。试求R_1和R_2的值，使得在铜-康铜热电偶两端温差为100℃时电路的输出为10mV，假设热电偶有常数值灵敏度43μV/℃。

3.5　与测量流量的孔板一起使用的差压传感器的输出是非线性的，输出电压与流量的平方成正比。试确定运算放大器信号调理器电路中反馈电路上的元件，使得输出线性化。

3.6　一个差分放大器的电压增益为100。如果两个输入电阻都为1kΩ，那么反馈电阻为多少？

3.7　一个差分放大器的电压差增益为2000，共模增益为0.2。共模抑制比是多少dB？

3.8　从传感器得到的数字信号被噪声污染，主要为干扰，且一般信号大小为数量级100V或更大。试阐述当这些信号作为输入时如何对微处理器提供保护。

3.9　一个铂电阻温度传感器在0℃时电阻为120Ω，它组成惠斯通电桥的一条臂。电桥其他臂电阻都为120Ω，在该温度下电桥平衡。铂电阻的电阻温度系数为0.0039/K。当温度变为20℃时电桥的输出电压为多少？输出端之间的负载可当作开路，提供给电桥电压的电源为6V，忽略其内阻。

93

3.10　膜片压力计采用4块应变片来测量膜片的位移。这4块工作应变片分别组成惠斯通电桥的4条臂，如图3-23所示。应变片灵敏度为2.1，电阻为120Ω。提供给膜片的差压使膜片同一侧的两块应变片受到1.0×10^{-5}的张力应变，而另一侧的两块则为1.0×10^{-5}的压缩应变。提供给电桥的电压为10V。电桥输出电压为多少？

3.11　惠斯通电桥在一条臂上有一个单应变片，其余臂上为与未受压时应变片的电阻相同的电阻。证明电桥的输出为$\frac{1}{4}V_sG\varepsilon$，其中$V_s$为电桥的电源电压、$G$为应变系数、$\varepsilon$为应变片的应变。

94

第4章 数字信号

本章目的

通过本章的学习，读者将能够：

- 解释模数转换和数模转换的原理和主要方法。
- 解释多路复用器的原理和使用方法。
- 解释数字信号处理的原理。

4.1 数字信号简介

大多数传感器是以模拟形式输出，其大小和输入的大小密切相关。当微处理器被用做测量或控制系统的一部分时，先把来自传感器的模拟输出转换成数字信号，然后才可以用来作为微处理器的输入。同样，大部分驱动器用模拟输入操作运行，所以从微处理器输出的数字信号必须先转换成模拟信号，然后才可以用做驱动器的输入。

4.1.1 二进制数

二进制仅有两个符号或者两个状态0和1，可以是0V或者5V的信号。这些数称为二进制数或者**位**。当某个数是由二进制来表示时，这个数中某个数字的位置表示每个数字的‘位权’，当我们从右向左计算的时候，‘位权’以2的倍数增加：

…	2^3	2^2	2^1	2^0
	位3	位2	位1	位0

例如，十进制数15可以表示为$2^0+2^1+2^2+2^3=1111B$。在一个二进制数中，0位称为是**最低有效位**（LSB），最高位称为**最高有效位**（MSB）。表示一个数字的位的组合称为字。因此1111是一个4位的字。这样的一个字可以用来表示信号的大小。8位称为**字节**。关于二进制数的更多讨论详见附录B。

4.2 模拟信号和数字信号

模数转换是把模拟信号转换成二进制字。图4-1a显示了模数转换的基本要素。

其过程是一个时钟发出周期性的脉冲信号给模数转换器（ADC），模数转换器每接收到一个时钟脉冲就对模拟信号进行一次采样。图4-1通过显示各个阶段相关的信号类型阐释了模拟数字转换。图4-1b是模拟信号，图4-1c是提供采样时间的时钟信号。采样的结果是一系列窄脉冲（见图4-1d）。**采样**和**保持**单元用来保存每个采样值直到下一个采样时钟脉冲到来，结果如图4-1e所示。采样和保持单元是必需的，因为模数转换器需要一定的时间把模拟信号转换成数字信号，称为**转换时间**。

对于3位二进制数的输出，图4-2说明了模数转换器的输出以及采样和保持电路的输入的关系。3位有$2^3=8$种可能的输出电平。因此，既然代表模拟输入的ADC的输出只可以是这8种可能电平中的一种，那么在一定范围的输入，输出不会改变。这8种可能的输出电平称为**量化电平**，模拟电压在两个相邻电平之间的差称为**量化区间**。因此对于图4-2给出的ADC，量化区间是1V。由于阶梯状性质的关系，所以数字输出并不是一直和模拟输入成正比，因此会有

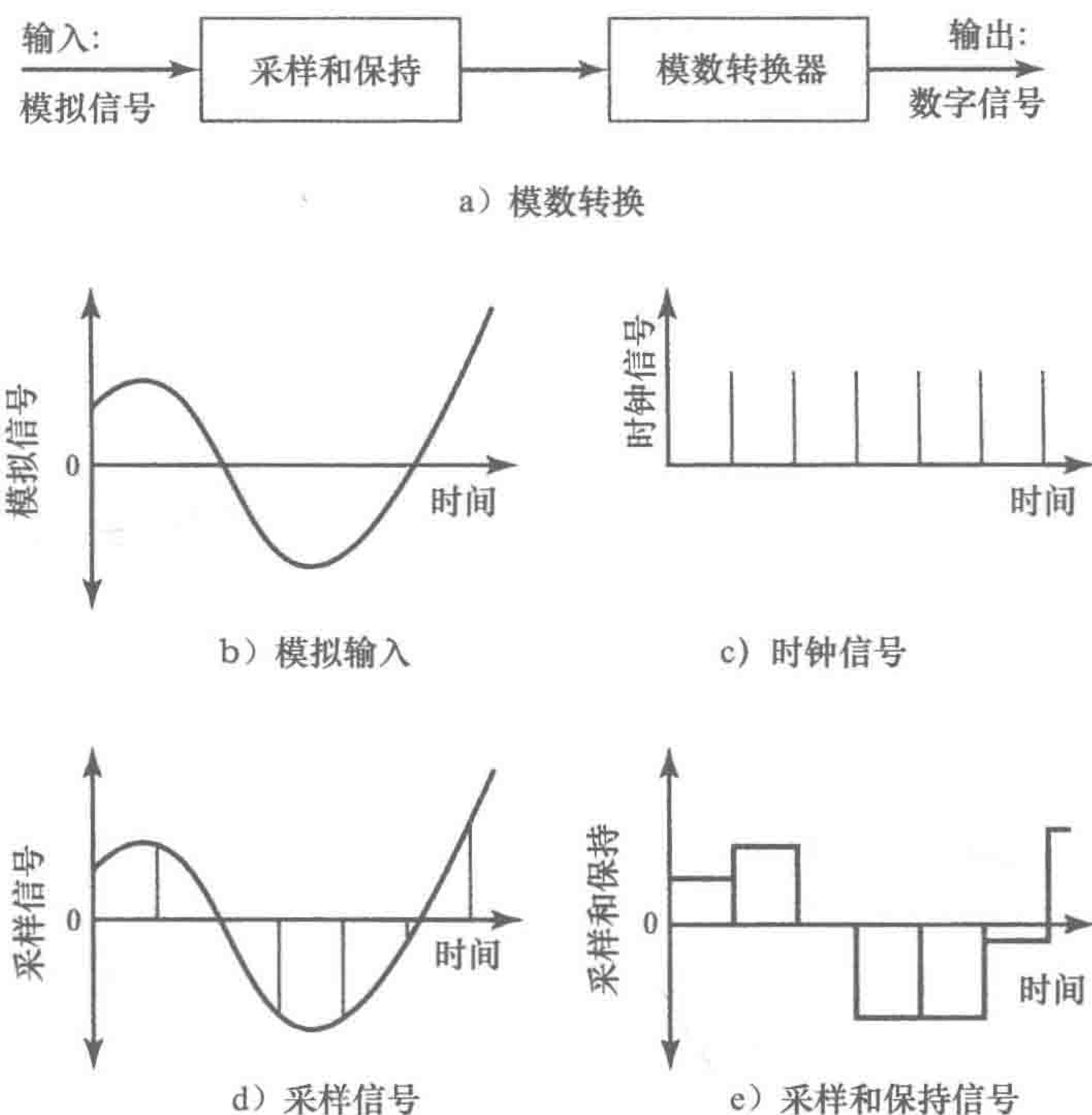


图 4-1　数模转换过程

误差，称为**量化误差**。当输入集中在量化区间时，量化误差为0，最大的量化误差等于量化区间的一半或者 $\pm\frac{1}{2}$位。

字长可能决定元件的**分辨率**，分辨率是导致数字输出改变的最小输入变化。数字输出的最小变化是在字中最低有效位的第1位，即最右边的位。因此一个有 n 位的字，满量程模拟输入 V_{FS} 被分为 2^n 份，输入中可以检测的最小的变化即分辨率是 $V_{FS}/2^n$。

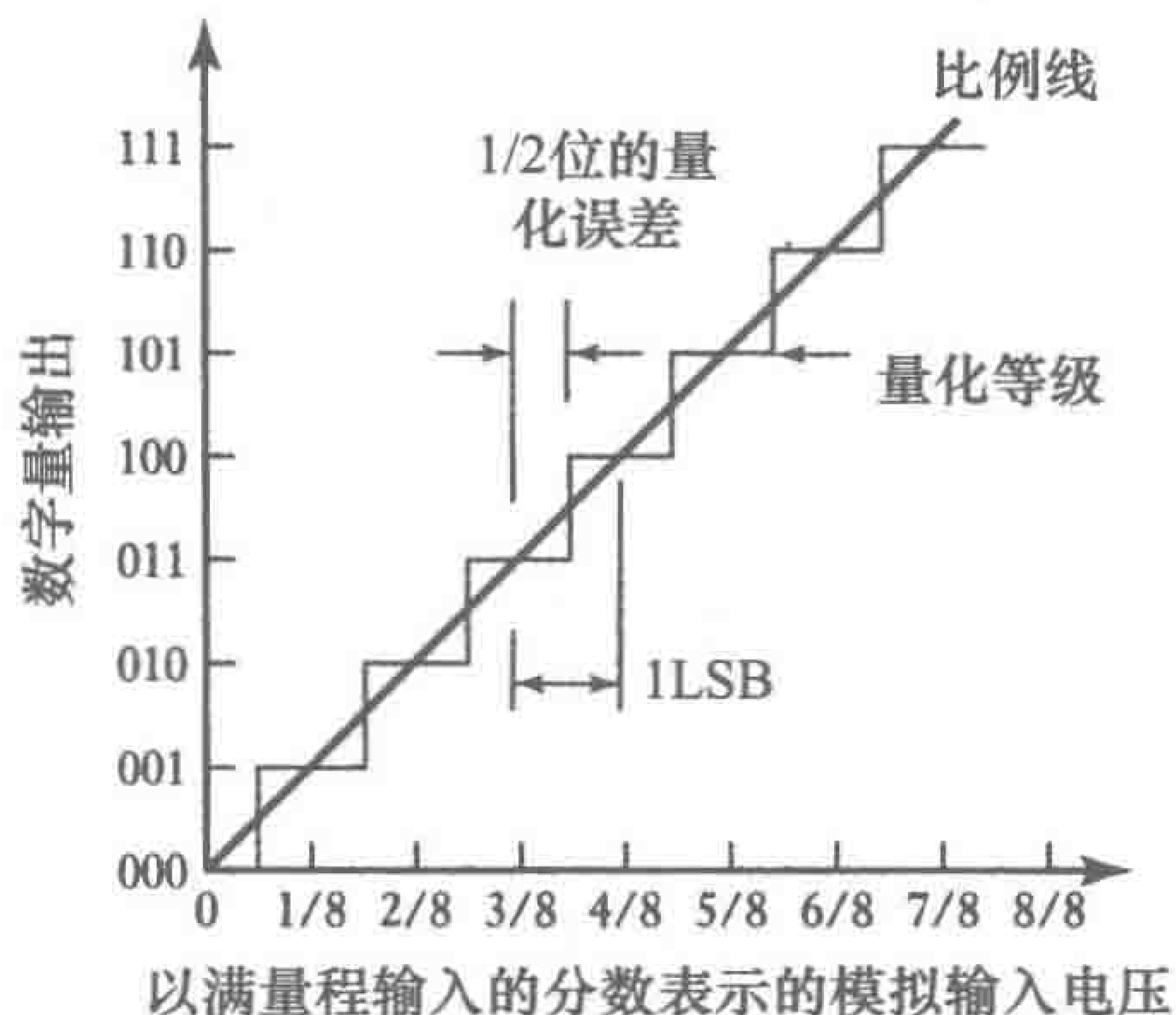


图 4-2　ADC 的输入/输出

如果有一个字长为10位，模拟信号输入范围为10V 的 ADC，那么用10位的字描述电压的种类个数是 $2^{10}=1024$，所以分辨率是 $10/1024=9.8$mV。

假设一个热电偶的灵敏度为0.5mV/℃。如果温度从0~200℃是以0.5℃为最小分辨率来测量，那么它通过 ADC 的输出所需要的字长是多少？传感器的满量程输出是 $200\times0.5=100$mV。对于一个字长为 n，电压会被分割成 $100/2^n$mV 份，分辨率为0.5℃的热电偶，必须有能力检测一个来自传感器的 $0.5\times0.5=0.25$mV 的信号。因此要求

$$0.25=\frac{100}{2^n}$$

可以求出 $n=8.6$。所以需要9位字长。

4.2.1 采样定理

ADC 每隔一段时间对模拟信号进行采样，然后将这些值转换成二进制数。那么应该要过多久对模拟信号采样一次才能使得输出结果可以代表模拟信号？

图 4-3 说明了相同的模拟信号使用不同的采样频率所带来的问题。当从样本重建信号时，只有当采样频率至少 2 倍于模拟信号中的最高频率时，样本才能给出信号的原始形式。这个准则称为**奈奎斯特（Nyquist）准则**或者**香农采样定理**。当采样频率低于最高频率的 2 倍时会得到错误的信号，这称为**失真**。图 4-3c 表示用一个远小于模拟信号频率的采样频率进行采样得到的模拟信号。

96 ~ 97

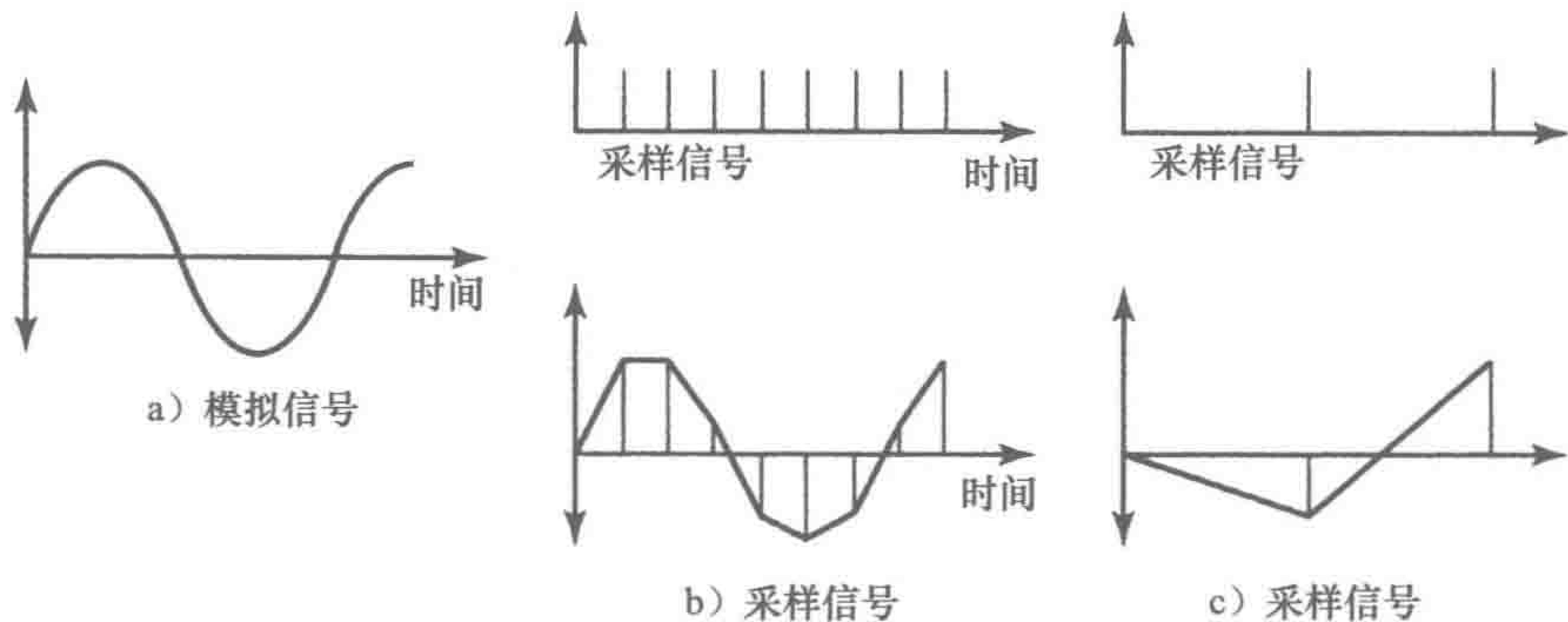


图 4-3 采样频率的影响

信号采样速度太慢，就会因低频失真而产生对高频信号分量的错误解析。高频噪声同样可以在转换过程中产生错误。为了减少由于失真和高频噪声带来的错误，在 ADC 之前使用一个低通滤波器，由于它的带宽使得只有低频信号才能通过，所以不会有失真错误。这样的滤波器称为**抗混叠滤波器**。

4.2.2 数模转换

数模转换器（DAC）的输入是二进制字，输出是代表该字非零位（non-zero bits）的加权和的模拟信号。因此，例如输入 0010 时的模拟输出必须是输入 0001 时模拟输出的 2 倍。图 4-4 表明了分辨率为 1V 的 DAC 的输入无符号二进制数时的模拟输出电压。每增加一位意味着输出电压增加 1V。

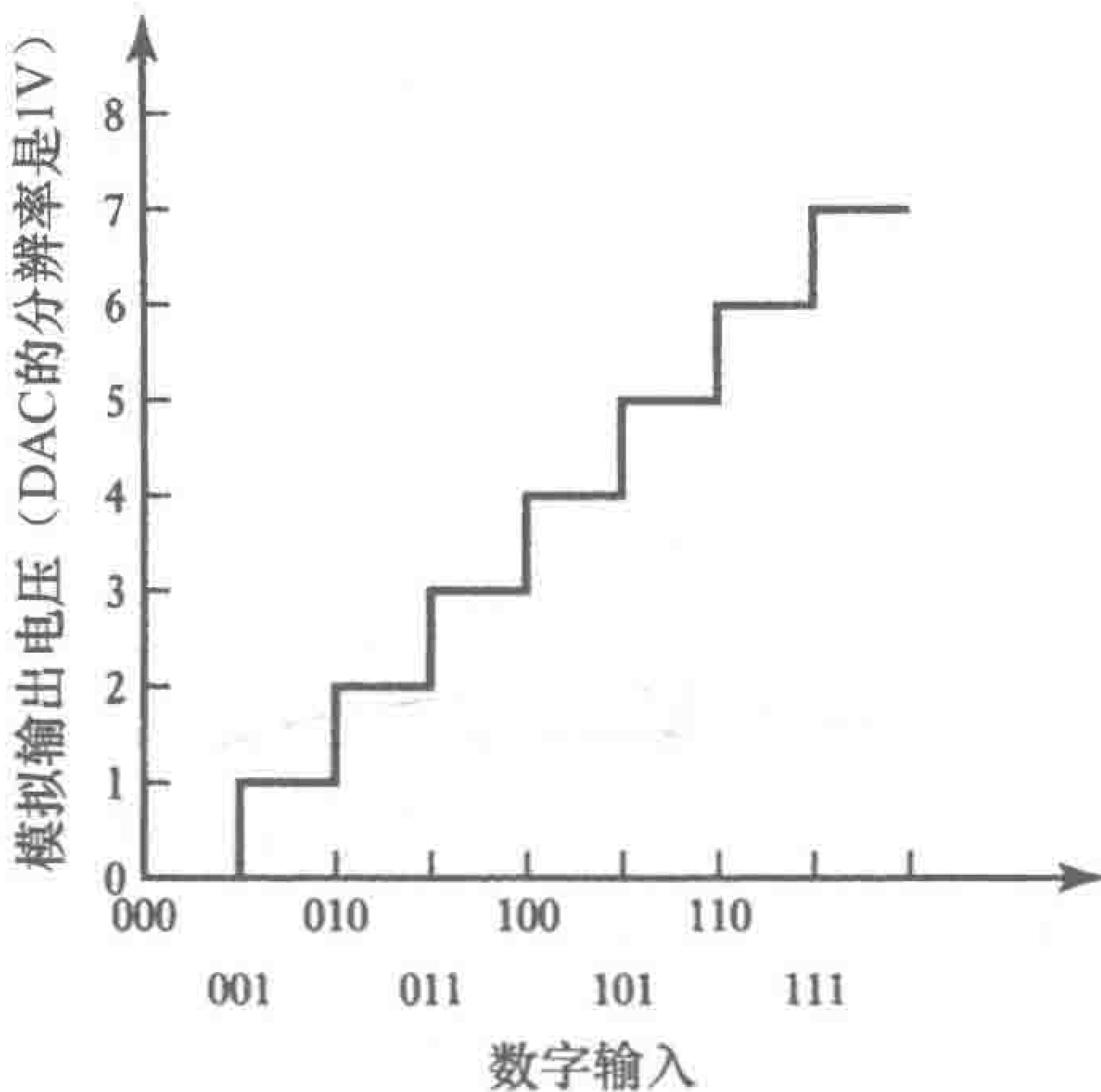


图 4-4 DAC 的输入/输出

考虑一个微处理器给出一个 8 位字的输出的情况。这是通过 8 位的 DAC 送入控制阀。控制阀需要 6V 的电压才能完全打开。如果完全打开状态用 11111111 表示，那么 1 位的变化对应阀的输出是什么？

6V 的满量程输出电压将分成 2^8 份。1 位的变化会引起输出电压 $6.0/2^8=0.023V$ 的变化。

98

4.3 数模转换器和模数转换器

下面是经常遇到的 DAC 和 ADC 的形式。

4.3.1 DAC

一种简单的 DAC 是用求和放大器（参见 3.2.3 节）求出所有输入字中非零位的加权

总和（见图4-5）。参考电压通过电子开关和电阻相连，对应一个二进制位。输入电阻值取决于开关相对应的位，从LSB开始的位相对应的电阻值依次减半。因此，电压的总和是一个字的所有位的加权总和。这样的系统称为**加权电阻网络**。运算放大器的作用是作为一个缓冲器，以确保电阻网络的输出电流不受输出负载的影响，也可以调节输出增益使电压范围适合于特定的应用。

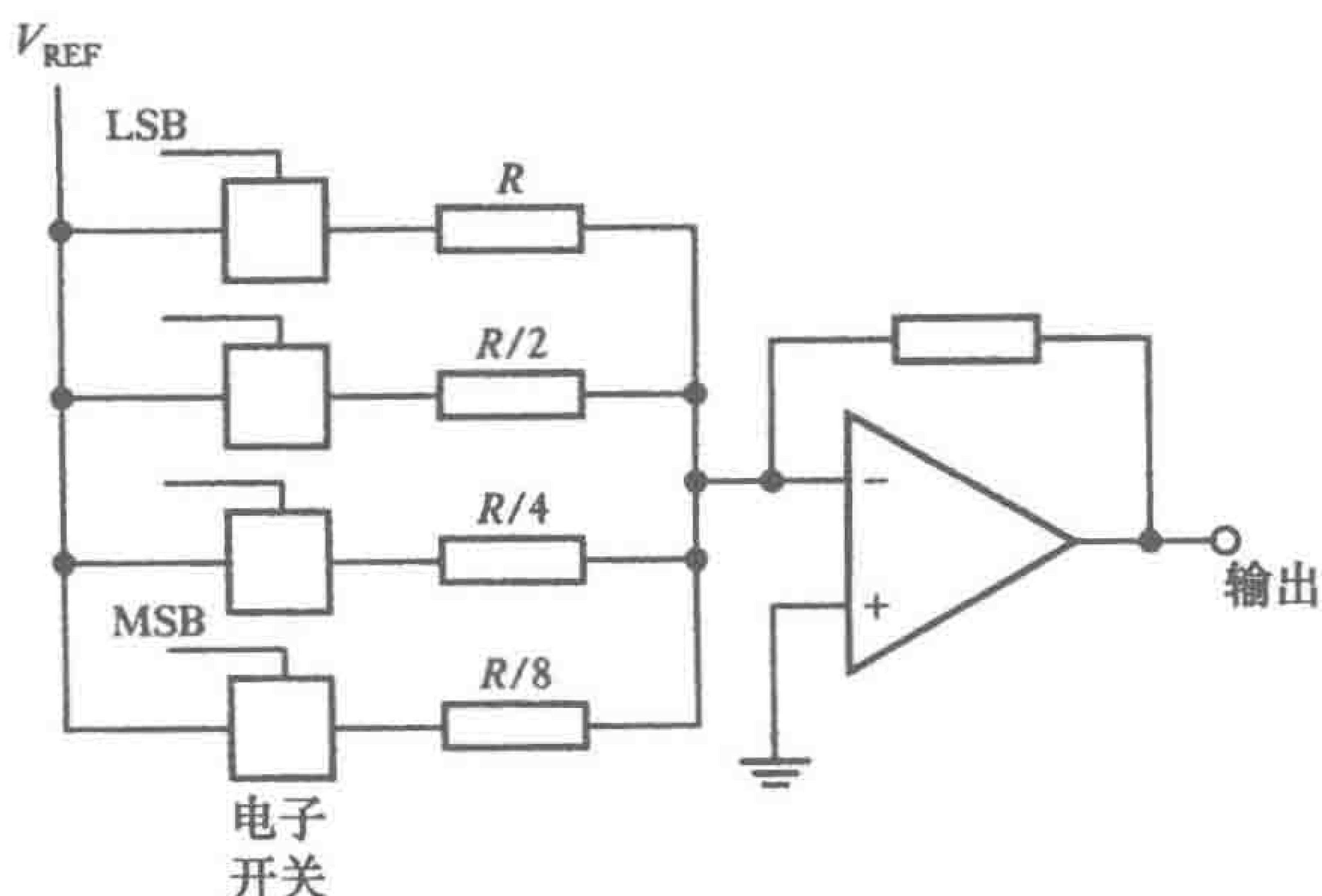


图4-5　加权电阻DAC

加权电阻网络的一个问题是每个电阻都必须是精密电阻，并且很难在广泛的电阻值范围内获得这样的精密电阻。因此这种形式的DAC转换器往往被限制到4位。

99

另一个更加常用的形式是使用***R*-2*R*梯形网络数模转换器**（见图4-6）。它克服了难以获得广泛范围内的精密电阻的问题，只需要两个电阻值。根据输入是1或0，通过梯形电阻网络的开关切换到参考电压或者0V来产生输出电压。

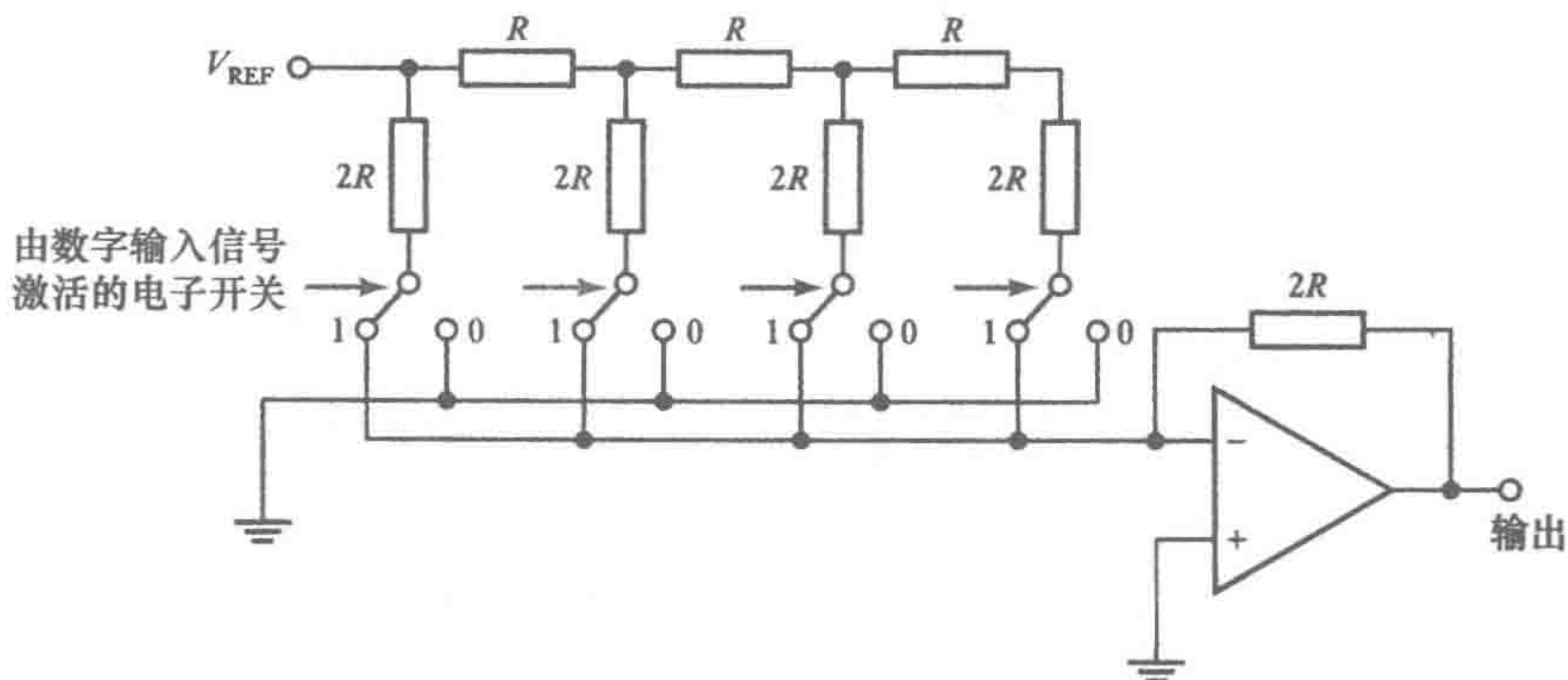


图4-6　*R*-2*R*梯形DAC

图4-7显示了GEC Plessey ZN558D 8位锁存输入DAC使用*R*-2*R*梯形网络的详细信息。转换完成后，8位结果存储在一个内部锁存器中，直到下一次转换完成。当ENABLE为高电平的时候，数据保留在锁存器中，当ENABLE为低电平的时候，锁存器清空。锁存器只是一个保留输出直到下一个新的输出代替它的器件。有锁存器的DAC可以直接连接到微处理器的数据总线，只是把它视为一个发送数据的地址。没有锁存器的DAC可以连接到外围接口适配器（PIA），这种器件提供锁存功能（见13.4节）。图4-8显示了当输出电压在0到参考电压之间变化时，ZN558D如何与微处理器一起使用，这称为**单极操作**。如果$V_{ref.in}$ = 2.5V，当R_1 = 8kΩ，R_2 = 8kΩ时输出范围是+5V；当R_1 = 16kΩ，R_2 = 5.33kΩ时输出范围是+10V。

DAC的指标涉及下列术语：

1）满量程输出，即输入字上的所有位都是1时对应的输出。对于ZN558D，它通常是2.550V。

2）分辨率，8位DAC一般适用于大多数的微处理器控制系统，ZN558D是8位。

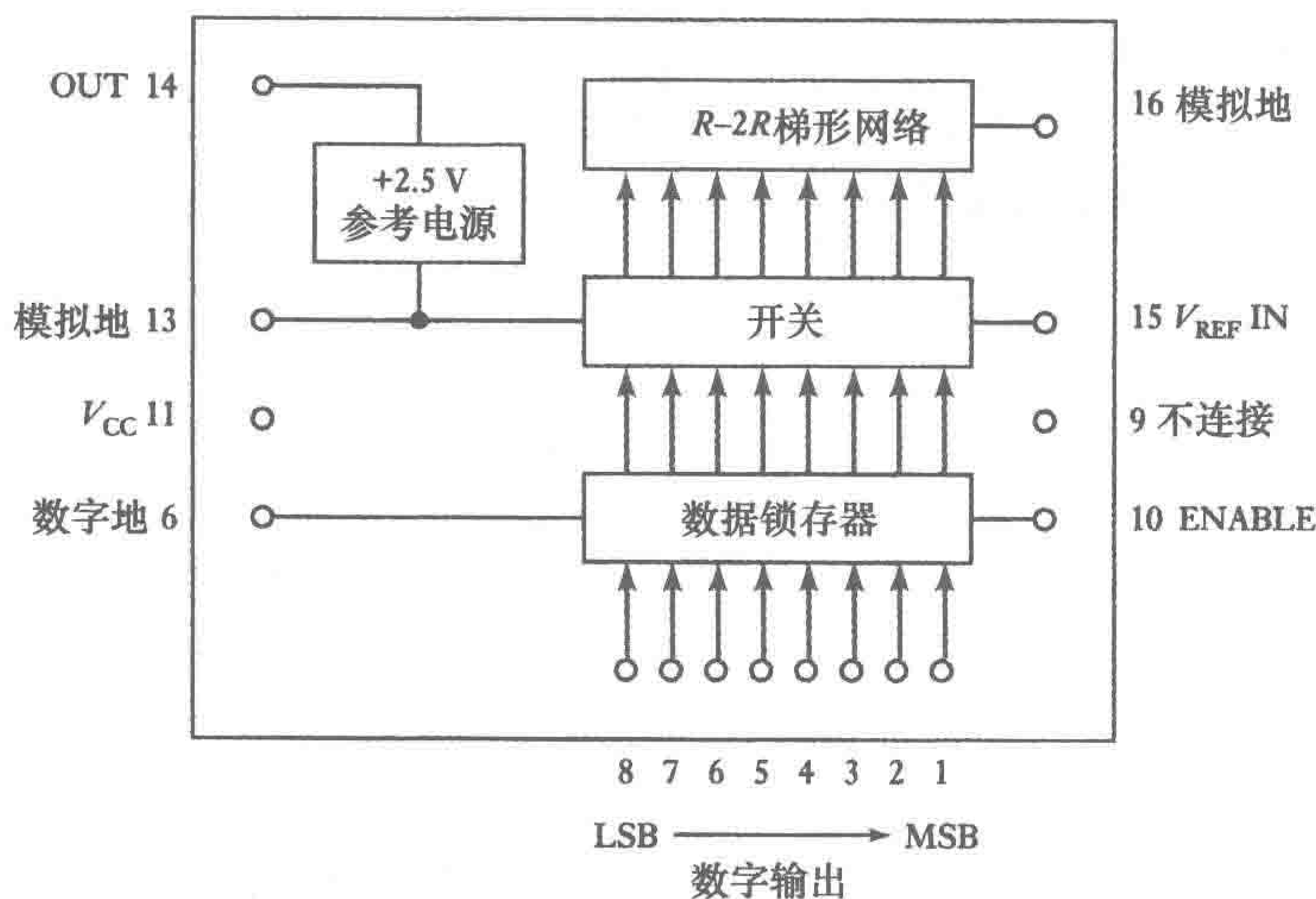


图 4-7　ZN558D DAC

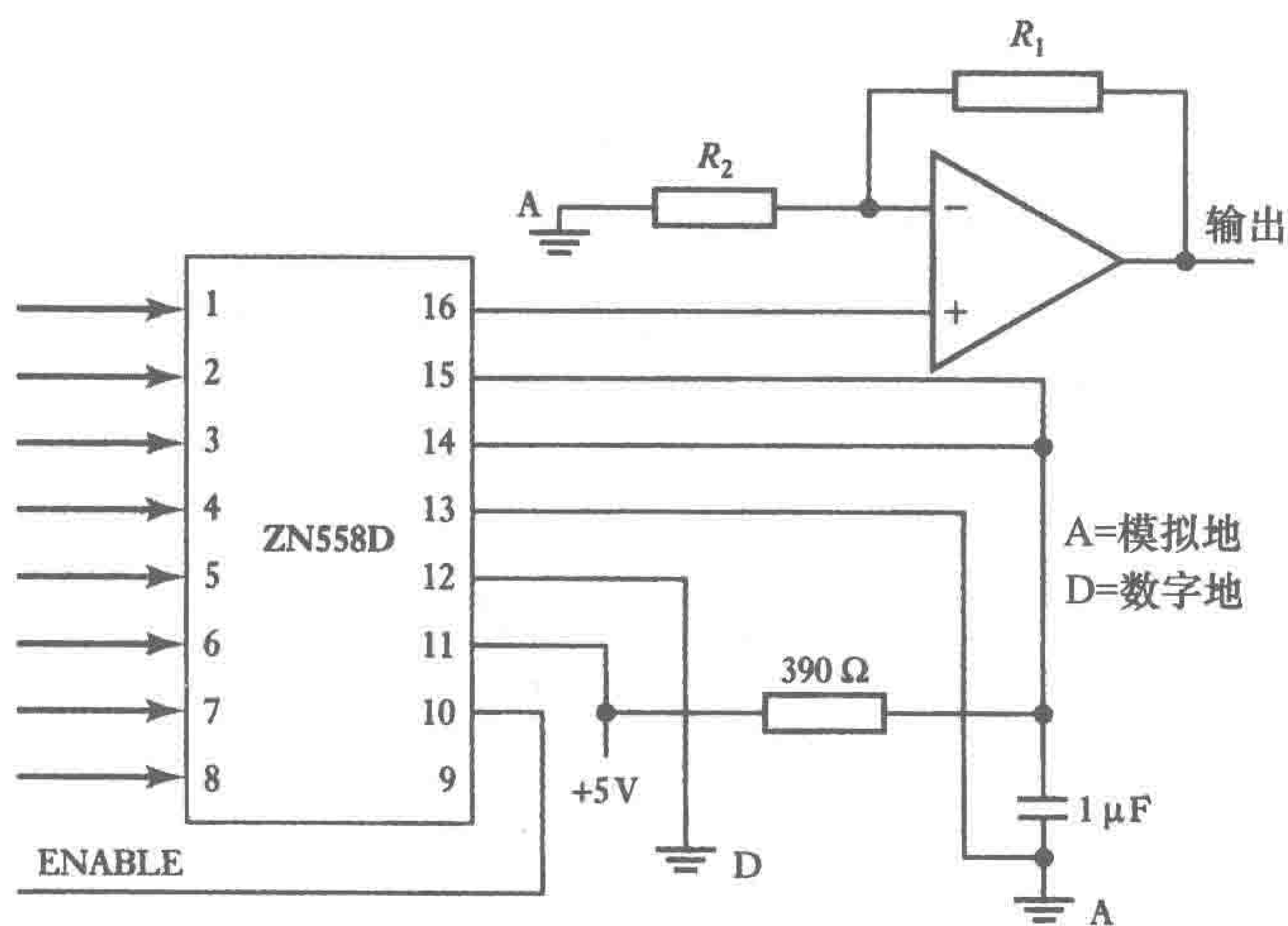


图 4-8　单极性操作

3）建立时间，这是在二进制数变化后 DAC 达到新电压的 ±0.5LSB 范围内所用的时间。ZN558D 的建立时间是 800ns。

4）线性度，这是通过 0 和满量程输出的直线的最大偏差。ZN558D 最大偏差是 ±0.5LSB。

4.3.2　ADC

ADC 的输入是模拟信号，输出是表示输入信号电平的二进制字。ADC 的种类很多，最常见的是逐次逼近式（successive approximation）ADC、斜坡式（ramp）ADC、双斜积分式（dual ramp）ADC 和快闪式（flash）ADC。

逐次逼近是最常用的方法。图 4-9 说明了一种相关的子系统。对一个时钟产生的周期性序列脉冲以二进制的方式进行计数，然后把该二进制数通过 DAC 转换成模拟电压。电压逐步上升，与传感器的模拟输入电压比较。

当时钟产生的电压超过输入模拟电压时，门电路关闭，时钟脉冲停止计数。这时，计数器的输出就成为模拟电压的数字信号。然而我们可以从 1 开始比较（LSB），然后逐位增加。一种更快的方法是逐次逼近。这就要选择小于模拟值的 MSB，然后加入连续的总量不超过模拟值

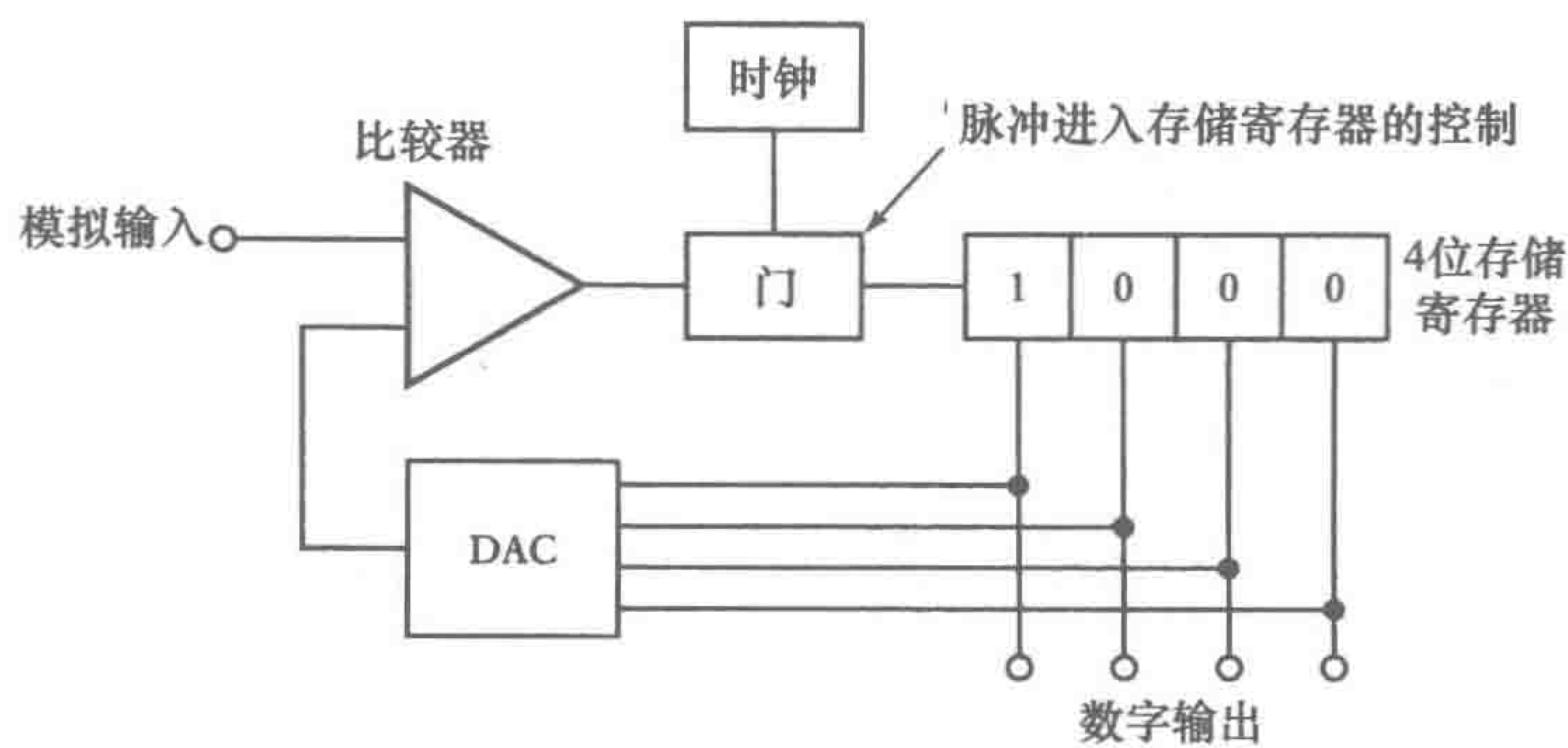


图 4-9 逐次逼近 ADC

的较低位。例如，我们可以从 1000 开始比较，如果太大，我们尝试 0100。如果太小了，我们尝试 0110。如果还是过大，我们尝试 0101。因为要按顺序尝试数中每一位，对于一个 n 位的数需要进行 n 步才能做出比较，所以如果时钟频率为 f，那么脉冲之间的时间间隔是 $1/f$。因此产生数的时间（即转换时间）就是 n/f。

100~101

图 4-10 显示了用逐次逼近方法设计的用于微处理器的 8 位 ADC（GEC Plessey ZN439）的典型形式。图 4-11 显示了如何连接才能使得它被微处理器控制，以及如何发送数字输出信号给微处理器。所有的有源电路，包括时钟，都集成在一个芯片上。把片选（chip select）引脚置为低电平，首先选择 ADC。当开始转换（start conversion）引脚收到一个负脉冲时转换开始。转换结束时状态（Status）引脚变为低电平。将数字输出信号发送到内部缓冲器存储，直到输出使能（output enable）引脚为低电平时开始读取。

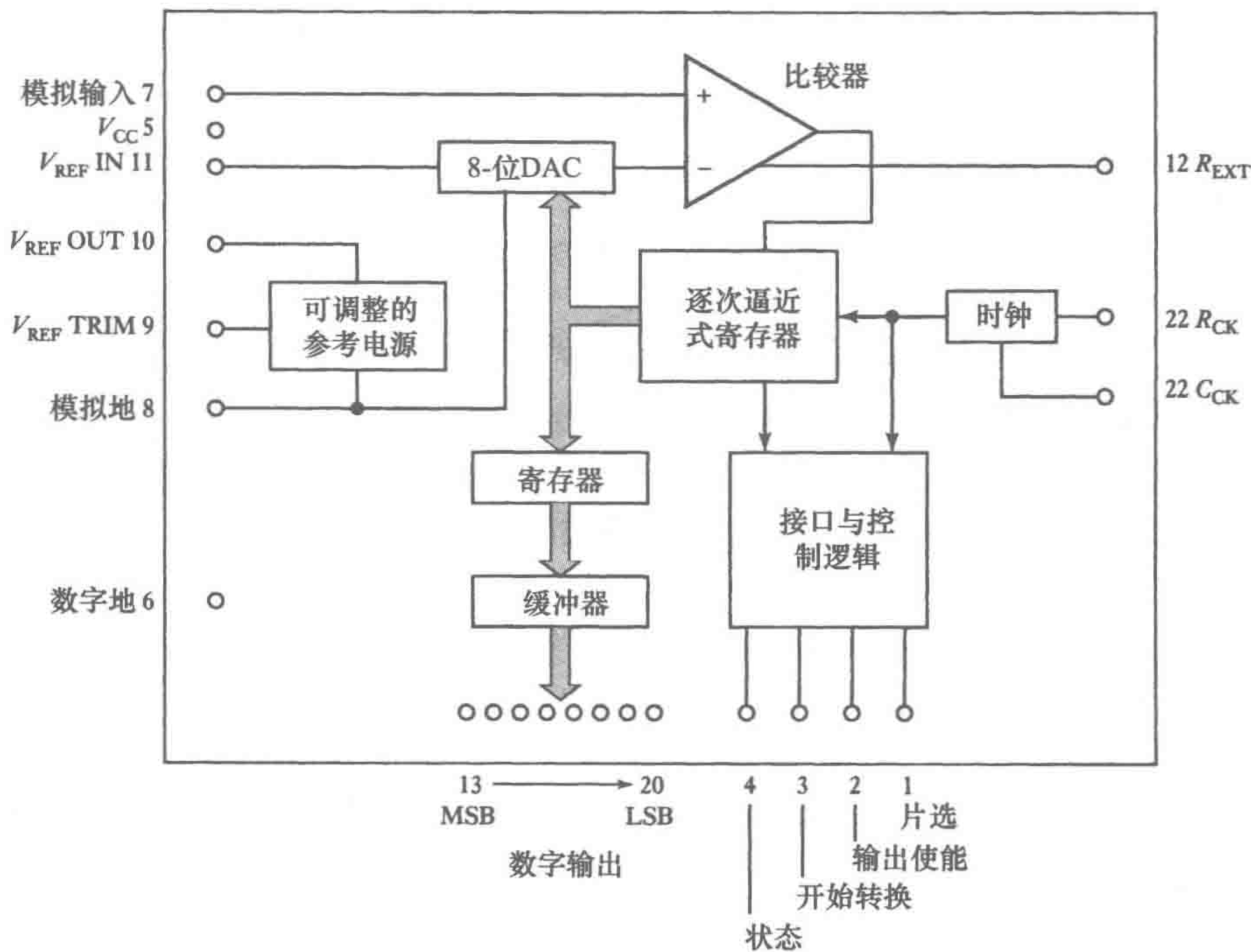


图 4-10 ZN439 ADC

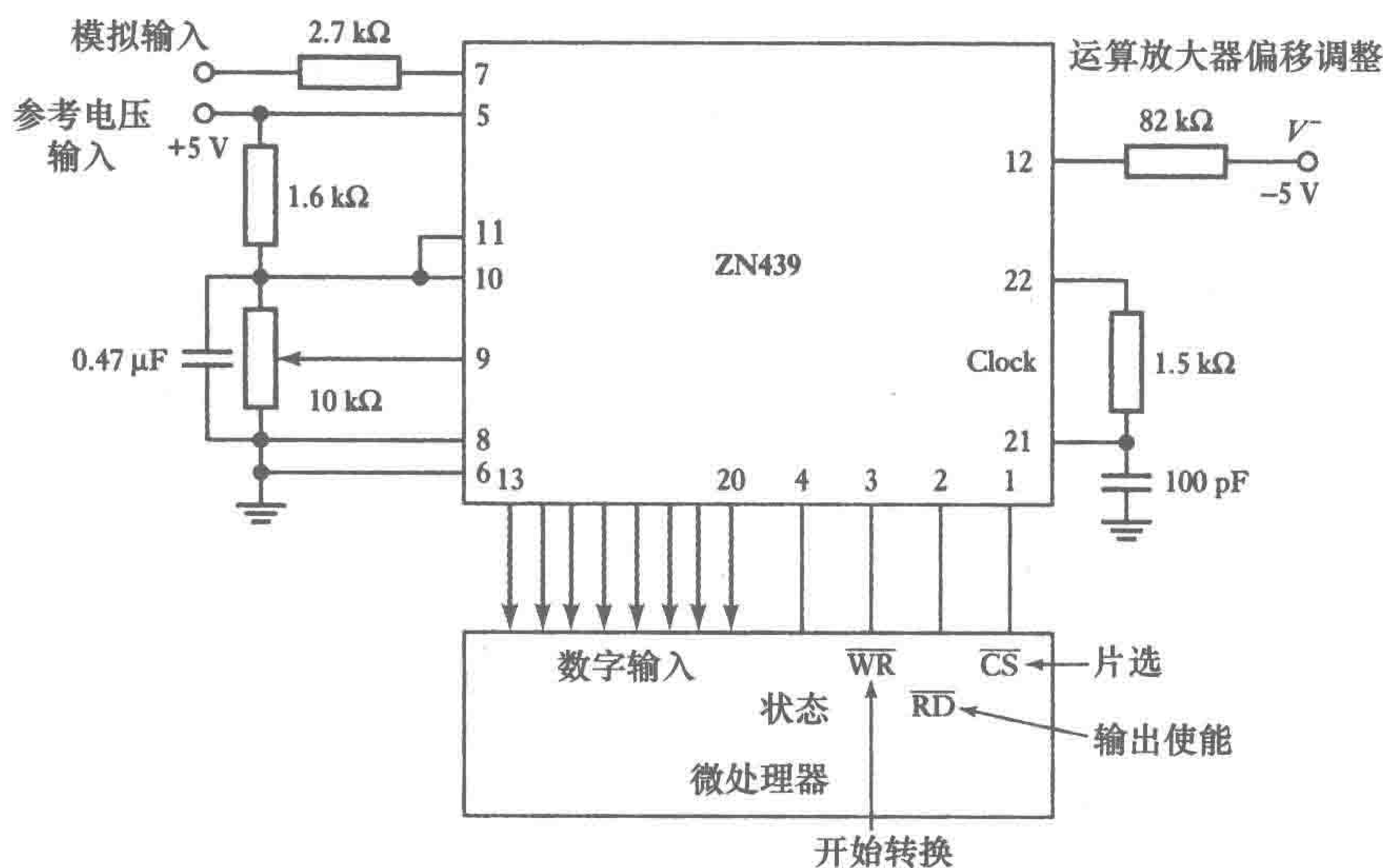


图 4-11 与微处理器相连的 ZN439

斜坡式 ADC 包括一个模拟电压，它以一个恒定的速率递增，即所谓的斜坡电压，通过比较器和来自于传感器的模拟电压相比较。斜坡电压增加到传感器电压所花费的时间取决于采样的模拟电压的大小。斜坡电压启动时，门电路打开，二进制计数器对时钟脉冲计数。当两个电压相等时，门电路关闭，该计数值就是采样的模拟电压的数字表示。图 4-12 表示了一个斜坡式 ADC 的子系统。

102

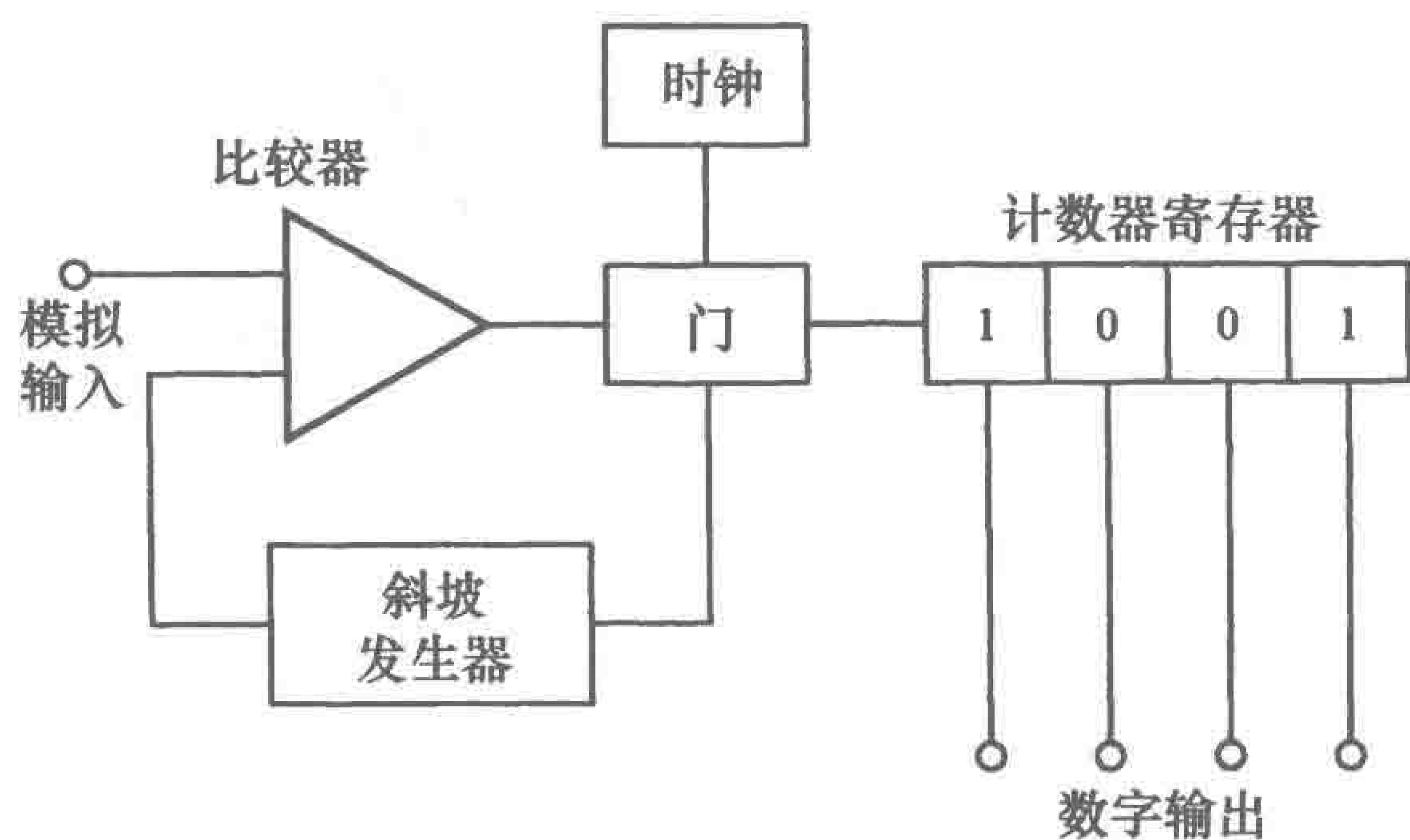


图 4-12 斜坡式 ADC

双斜积分转换器比斜坡式 ADC 更常见。图 4-13 显示了基本的电路。模拟电压施加到一个驱动比较器的积分器（integrator）上。只要积分器的输出是几毫伏以上，比较器就输出高电平。当比较器输出高电平时，脉冲通过 AND 门给二进制计数器。计数器对脉冲计数，直到产生溢出。然后计数器复位为 0，发送信号给开关断开未知电压的连接，并和参考电压相连，然后重新开始计数。参考电压和输入电压的极性相反。积分器的电压以一个和输入电压成正比的速率降低。当积分电压输出为 0 时，比较器变为低电平，给 AND 门输入低电平，所以时钟关闭。于是，该计数值就是这个输入模拟电压的度量。双积分 ADC 具有良好的噪声抑制作用，因为积分行为平均了采样周期内随机的积极和消极的影响，然而这种 AD 转换的速率是非常低的。

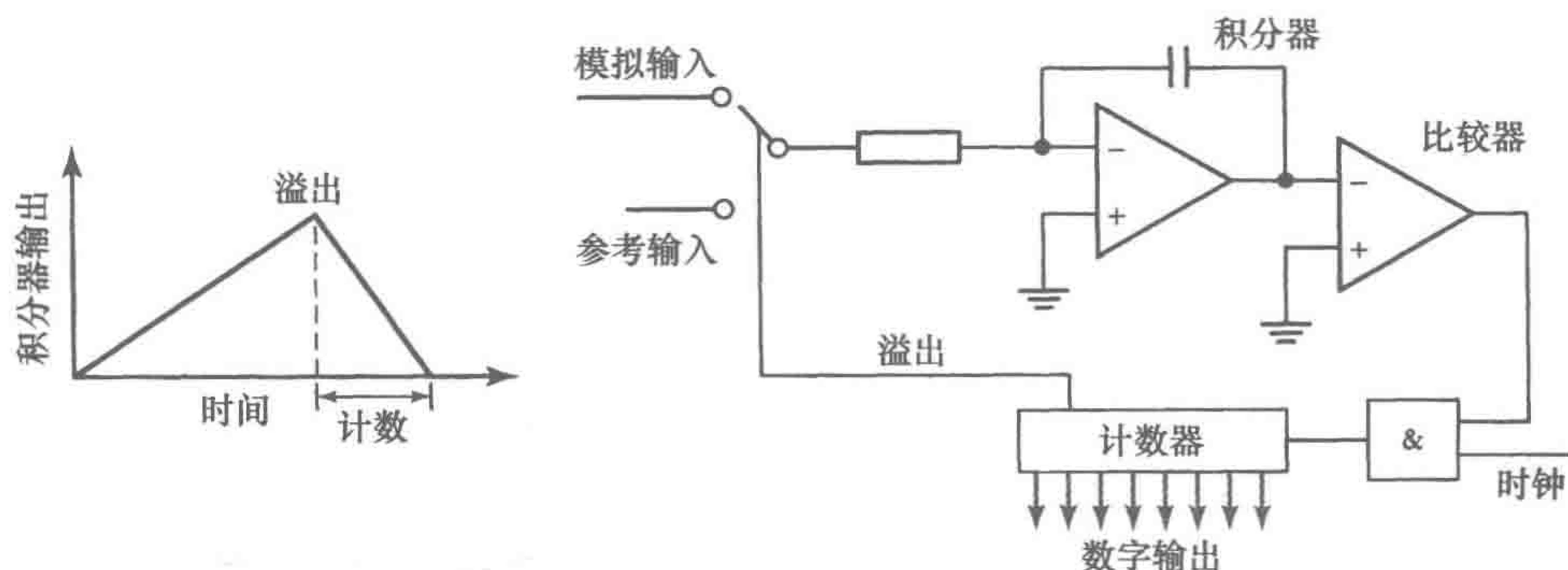


图 4-13　双斜积分式 ADC

快闪式 ADC 是非常快的。对于一个 n 位转换器，2^n-1 个独立的电压比较器并联使用，把每个模拟输入电压作为一个输入（见图 4-14）。每一个比较器的参考电压（通过梯形电阻施加）都比下一个参考电压高出 1 位所代表的电压值。因此，当模拟电压施加到 ADC 时，所有这些模拟电压大于参考电压的比较器都有高的输出，反之则有低的输出。所产生的输出并行送入逻辑门系统，转换成一个数字。

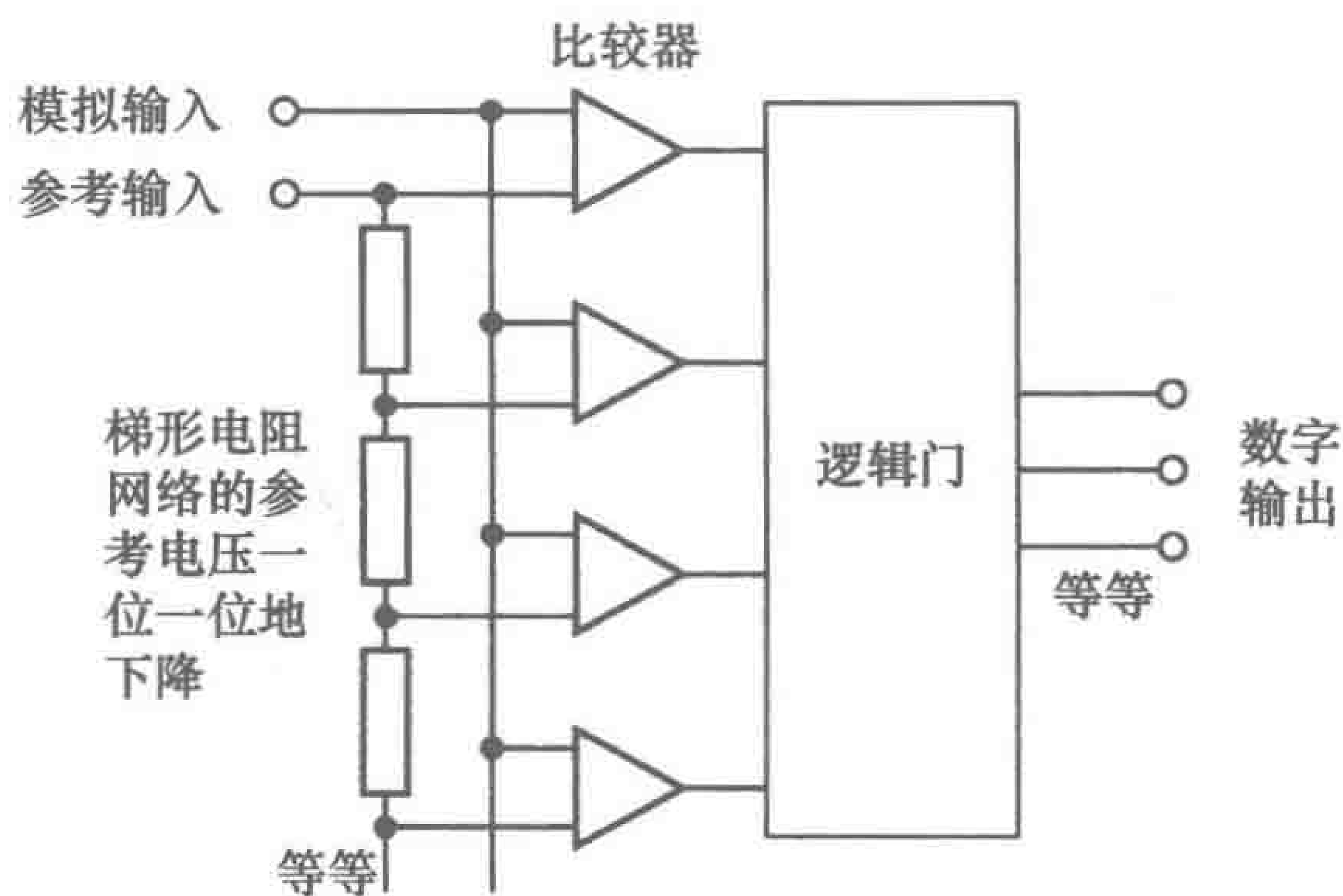


图 4-14　快闪式 ADC

在考虑 ADC 的指标时，会遇到下列术语：

1）转换时间，即完成输入信号转换所需的时间。它确定不失真采样信号频率的上界，最大频率是 1/(2 × 转换时间)。

2）分辨率，将满量程信号分成 2^n 份，n 是位数。它经常只是由位数指定。

3）线性误差，这是通过 0 到满量程输出的直线的偏差。最大偏差是 ±0.5LSB。

表 4-1 显示了常用 ADC 的指标。

表 4-1　ADC

ADC	类型	分辨率（位）	转换时间（ns）	线性误差（LSB）
ZN439	SA	8	5000	±1/2
ZN448E	SA	8	9000	±1/2
ADS7806	SA	12	20 000	±1/2
ADS7078C	SA	16	20 000	±1/2
ADC302	F	8	20	±1/2

注：SA = 逐次逼近式，F = 快闪式。

103～104

4.3.3　采样和保持放大器

ADC 把模拟信号转换成数字信号需要一定的时间，如果在转换期间模拟信号发生变化，则会出现问题。为了克服可能出现的问题，使用采样和保持系统采样模拟信号，当转换开始时保持模拟信号。

基本电路（见图4-15）包括采样用的电子开关、保持用的电容以及用运算放大器设计的电压跟随器。电子开关受到控制，这样采样是在控制输入的瞬间决定的。当开关关闭时，输入电压施加到电容两端，输出电压和输入电压相同。当开关关闭时，如果输入电压改变，则电容两端的电压和输出电压也相应地改变。当开关打开时，电容保持其电量，输出电压和开关断开瞬间的输入电压是相等的。因此电压保持直到开关再次关闭。电容器需要充电达到新的输入模拟电压的时间称为**采样时间**，它取决于开关关闭时电容的大小和电路的电阻值。典型值是4μs数量级。

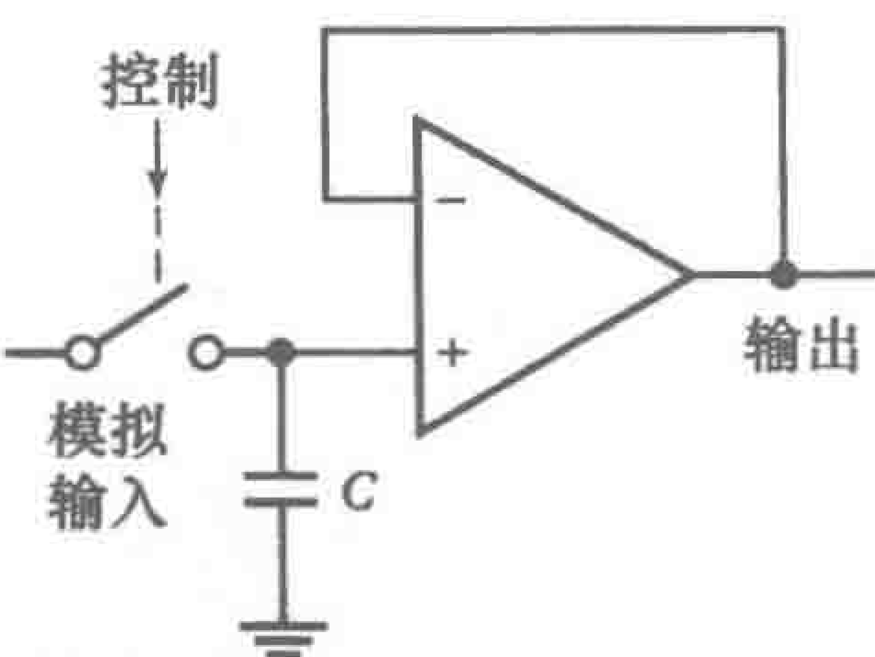


图4-15 采样与保持

4.4 多路复用器

多路复用器是一个能够获得多个数据输入来源的电路，它选择其中一个输入通道的数据作为输出。实际应用过程中需要对多个不同的位置进行测量，因此不是针对每一个量对应地用分开的ADC和微处理器来独立测量，而是用多路复用器依次选择每一个测量输入并分别切换到一个ADC和微处理器来轮流进行测量（见图4-16）。多路复用器实质上就是一个电子开关器件，使每个输入能够依次被采样。

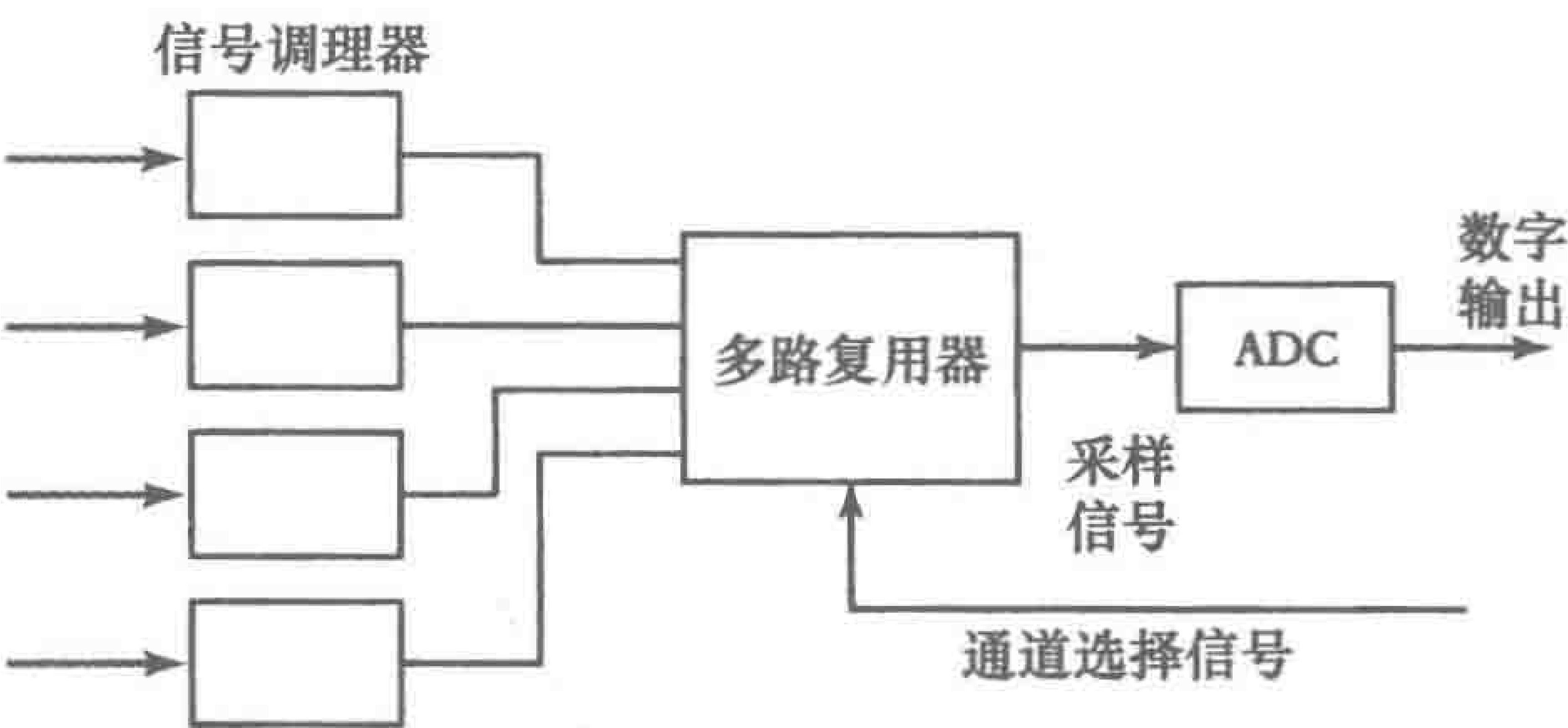


图4-16 多路复用器

现有的模拟多路复用器（如DG508ACJ）有8个输入通道，每个通道有3位二进制地址用于选择。采样之间的转换时间是0.6μs。

4.4.1 数字多路复用器

图4-17显示了一个可以用来选择数字数据输入的多路复用器的基本原理。为了简单起见，只有两个输入通道系统。选择输入的逻辑电平决定哪个AND门打开，使数据输入通过OR门到达输出（参考第5章关于这些门的讨论）。集成封装的电路提供了不同种类的复用器，151型多路开关芯片可使8根线中有1根线被选择。153型芯片是在一块集成芯片上有两个4选1数据选择器，即4个输入中有一根线被选中提供数据。157型芯片是在一块集成芯片上有4个2选1数据选择器，即2个输入中有一根线被选中提供数据。 105

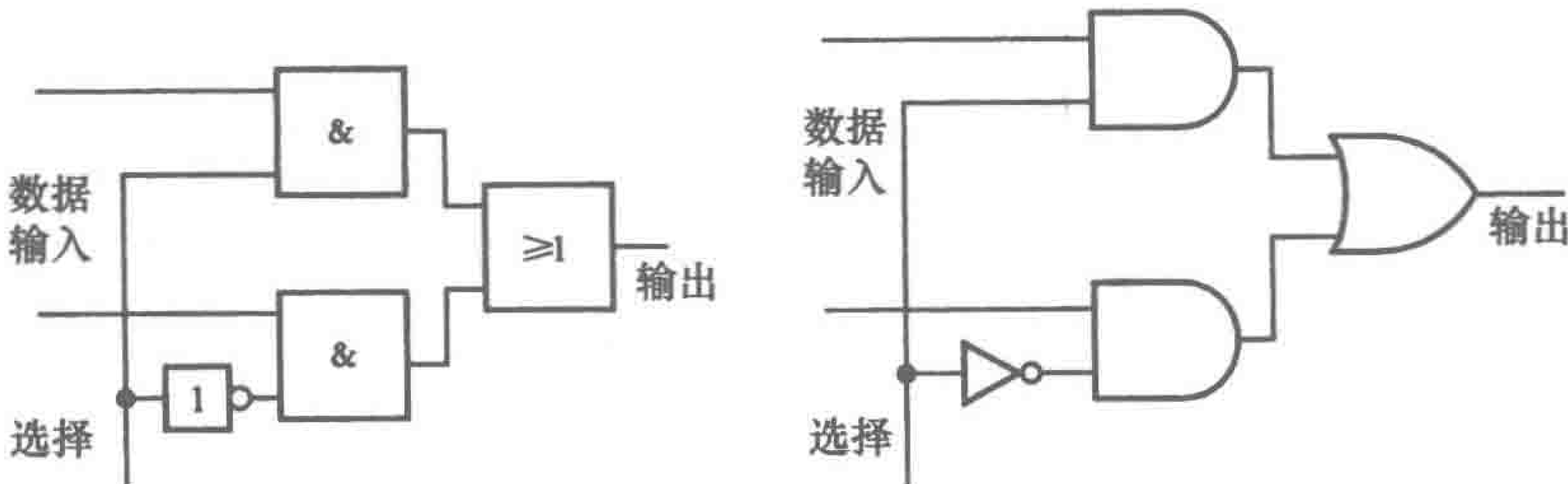


图4-17 双通道多路复用器

4.4.2 时分多路复用

经常需要一些外围设备共用一个微处理器的输入/输出线。所以每个外设可以提供不同的数据，数据传输过程中分配特定的时间片段是非常必要的。这称为**时分多路复用**。图 4-18 说明了如何用这种方法来驱动两个显示设备。图 4-18a 不是时分多路复用，但图 4-18b 是时分多路复用。

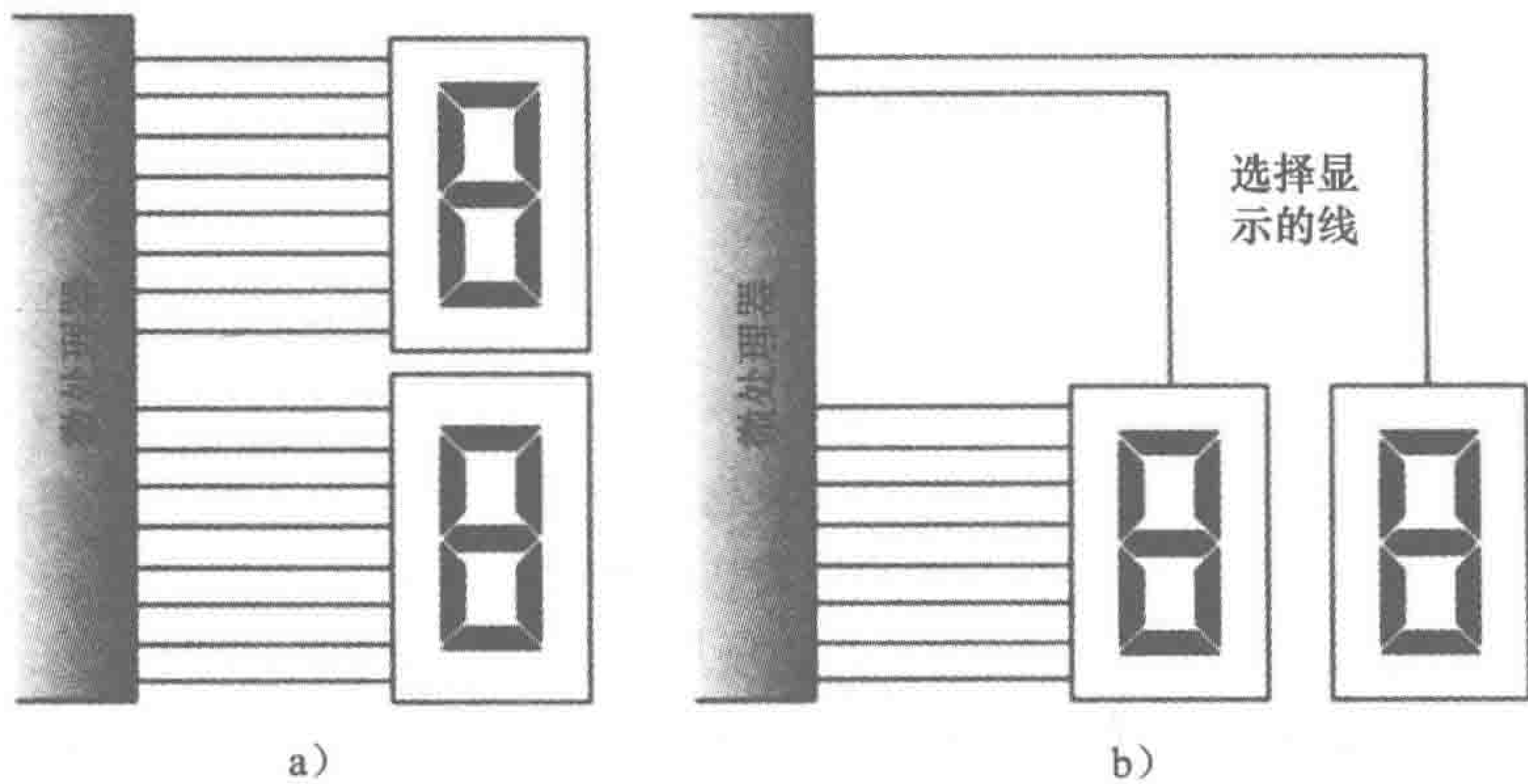


图 4-18　时分多路复用

4.5 数据采集

数据采集（DAQ）是用来从传感器采集数据，然后把数据输入到计算机中进行处理的过程。传感器一般要经过信号调理后，连接到插入计算机背面的数据采集板。DAQ 板是印制电路板，对于模拟输入，一般提供一个多路复用器、放大器、AD 转换器、寄存器和控制电路，以便于把采样的数字信号应用到计算机系统中。图 4-19b 显示了这样一个电路板的基本单元。

106

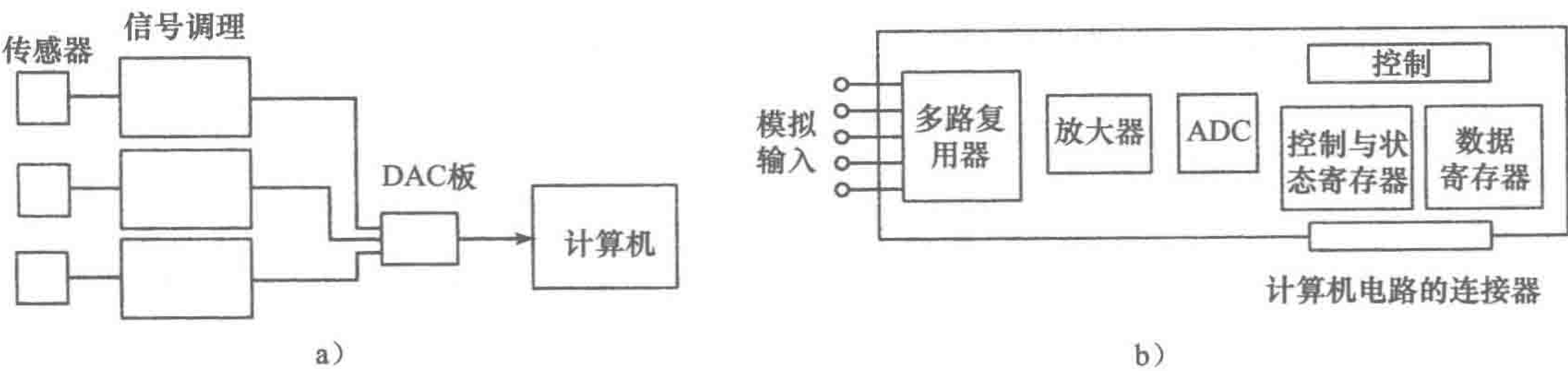


图 4-19　数据采集系统

计算机软件是通过 DAQ 板来控制数据采集。当程序需要从特定的传感器输入数据时，通过给控制和状态寄存器发送控制字来激活电路板。这样的“字”表示电路板必须进行的操作类型。因此电路板的多路复用器被切换到相应的输入通道。传感器的输入连接到该输入通道，然后通过一个放大器连到 ADC。AD 转换后所产生的数字信号被传递到数据寄存器，控制和状态寄存器中字的变化表明信号的到来。接着，计算机发出一个读取数据的信号，然后把数据读入计算机进行处理。必须确保采集数据时，计算机没有做任何其他工作。数据采集完成后，把信号传送到计算机，然后计算机中断所有正在执行的程序，从 DAQ 中读取数据后继续执行该程序。把数据传输到存储器的更快方法是不涉及计算机，而是直接把所采集的数据从电路板转

移到内存，这称为**直接内存寻址**（Direct Memory Address，DMA）。

数据采集板的指标包括模拟输入的采样速率，这可能是100kS/s（每秒100 000个样本）。采样的奈奎斯特准则表明，这种板可以采样的模拟信号的最大频率为50kHz，采样频率必须是最大频率分量的2倍。数据采集板除了上述的基本功能外，还可以提供模拟输出、定时器和计数器，可以用来提供传感器系统的触发器。

作为一个低成本多功能板用于IBM计算机的例子，图4-20显示了美国国家仪器公司的DAQ板PC-LPM-16的基本结构。此板有16个模拟输入通道，采样频率为50kS/s，一个8位数字输入和一个8位数字输出，可以有计数器/定时器输出。对多个通道按顺序扫描，从每个通道依次读取，也可以对一个通道进行连续扫描。

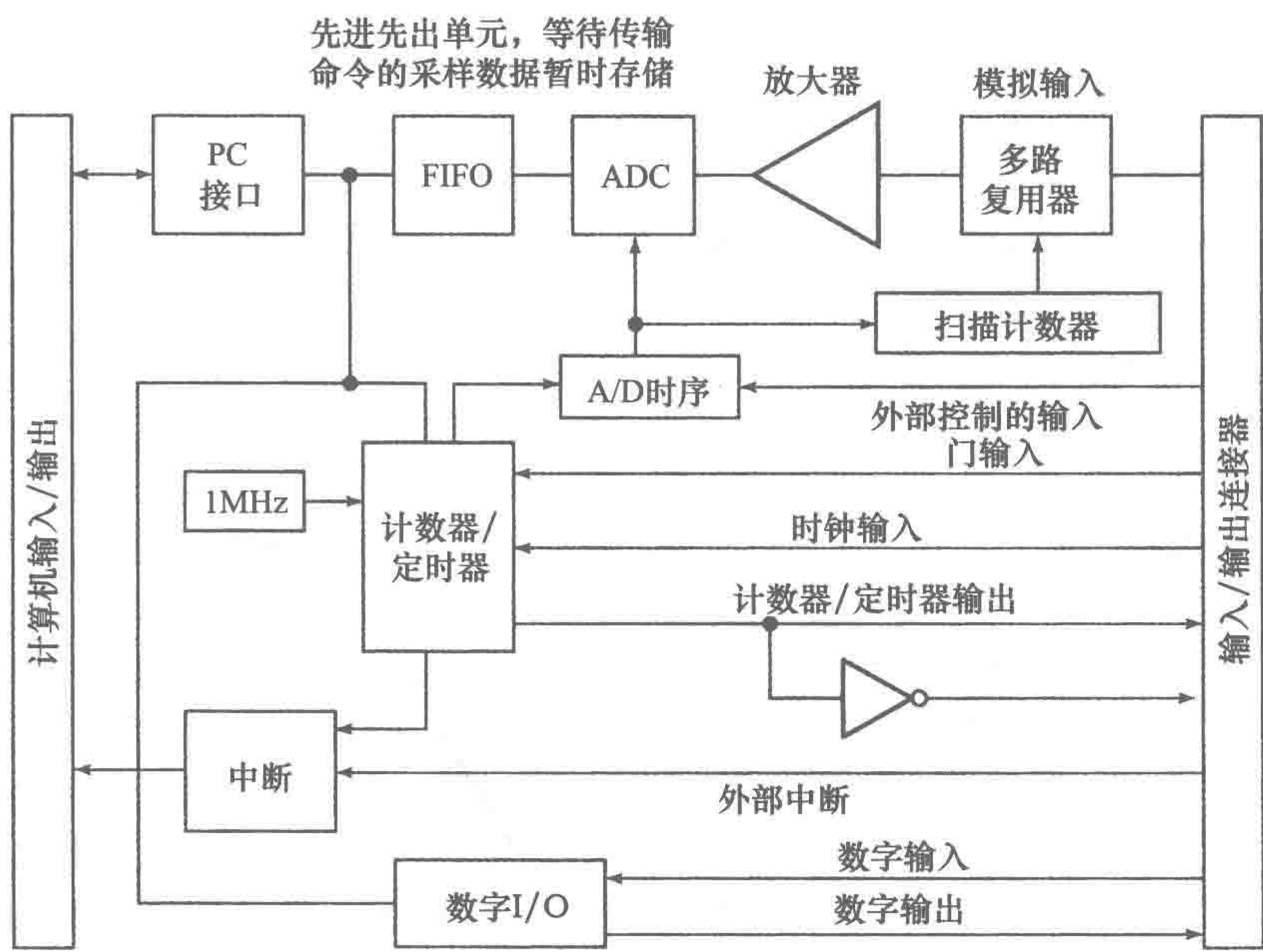


图4-20 PC-LPM-16 DAQ板

4.5.1 数据精度

数字信号处理的一个优点是，使用两种电压范围而不是两个确切的电压值来区分两者之间每位的二进制状态。因此可以减少噪声、漂移、元件容差等因素造成的电压波动对数据精度的影响，这是作为模拟电压传输的关键。例如，在5V系统中，两个二进制状态之间的差通常是3V。这样，两个信号可能是0V和5V或者1V和4V，仍然可以区分为0和1。

4.5.2 错误检测的奇偶校验方法

当数字数据从一个位置传送到另一个位置时，可能会因为传输过程中的电噪声使接收器不能接收来自发送器的相同信号，从而发生传输错误。有时噪声脉冲可能在某些点大到足以改变信号的逻辑电平。例如传输序列是1001，可是却收到1101。为了检测这种错误，通常用**奇偶校验位**。奇偶校验位是在传输代码组中一个额外的0或1。在**偶校验**时，选择位的值使得代码组中的总数（包括奇偶校验位）是偶数。例如传输1001时使用的校验位0给01001，则有偶数个1。在传输1101时使用的校验位1给11101，因此有偶数个1。使用**奇校验**时，校验位的选择使所有1的总数，包括校验位，是奇数。于是，如果一个代码组中接收器1的个数不是要求的奇偶数，则接收器知道有一个错误发生，将请求代码组重新传输。

奇偶校验的一个扩展是**求和校验**，通过发送一系列代表代码块的“二进制和”的位，可

107~108

校验代码块。奇偶校验与求和校验只能检测代码块中单一的错误，双错误不能被发现。而且错误不能被定位，所以接收器要进行矫正。目前已经设计出了定位错误的多错误检测技术和方法。

4.6 数字信号处理

数字信号处理或**离散时间信号处理**是微处理器用于处理信号的两个术语。数字信号是离散的时间信号，它们只在离散的时间处存在，是关于时间的不连续函数。而模拟信号调理要求放大器和滤波器电路等元件；而数字信号调理可以通过应用于微处理器的程序（即进行信号处理）来进行。如要改变用于模拟信号的滤波器的特征，必须改变硬件元件。而要改变数字滤波器的特征，则必须改变软件，即改变给微处理器的指令程序。

对于一个数字信号处理系统，有一个输入字代表脉冲的大小，另一个字代表输出的大小。在一个特定的瞬间，输出脉冲可以根据当前的输入脉冲、前面的脉冲输入和可能的前面的系统输出进行计算。

例如，微处理器所使用的程序可能会读取当前的输入值再加上前面的输出值，给出新的输出。如果认为当前的输入是在输入脉冲序列的第 k 个输入脉冲，可以表示为 $x[k]$。脉冲序列中的第 k 个输出脉冲可以表示为 $y[k]$。前一个输出，即第（$k-1$）个脉冲，可以表示为 $y[k-1]$。因此，可以输出等于当前的输入加上前一个输出的程序：

$$y[k] = x[k] + y[k-1]$$

这样的方程称为**差分方程**。它给出了离散时间系统的输入和输出之间的关系，可以与描述随时间连续变化的系统输入和输出之间关系的微分方程媲美。对于上述差分方程，假设有一个采样正弦波信号输入，给出一个脉冲序列：

$$0.5, 1.0, 0.5, -0.5, -1.0, -0.5, 0.5, 1.0, \cdots$$

$k=1$ 的输入脉冲大小为 0.5。如果我们假定先前的输出是 0，即 $y[k-1]=0$，则 $y[1]=0.5+0=0.5$。$k=2$ 的输入脉冲大小为 1，于是 $y[2]=x[2]+y[2-1]=1.0+0.5=1.5$。$k=3$ 的输入脉冲大小为 0.5，于是 $y[3]=x[3]+y[3-1]=0.5+1.5=2.0$。$k=4$ 的输入脉冲大小为 -0.5，于是 $y[4]=x[4]+y[4-1]=-0.5+2.0=1.5$。$k=5$ 的输入脉冲大小为 -1.0，于是 $y[5]=x[5]+y[5-1]=-1.0+1.5=0.5$。因此输出脉冲是：

$$0.5, 1.5, 2.0, 1, 5, 0.5, \cdots$$

我们可以继续用这种方式求出所有的脉冲输出。

再比如，有一个差分方程

$$y[k] = x[k] + ay[k-1] - by[k-2]$$

109

输出是当前输入加上 a 倍的前一个输出，再减去 b 倍的倒数第二个输出。如果 $a=1$，$b=0.5$，考虑上述的采样正弦波信号输入，输出变成

$$0.5, 1.5, 1.75, 0.5, -1.37, \cdots$$

有一个差分方程，产生输出的方式与对连续时间信号积分类似。两个时间之间的连续时间信号的积分，可以认为是该时间区域上连续时间函数与时间所围成区域的面积。如果两个离散时间信号是 $x[k]$ 和 $x[k-1]$，则它们之间的时间间隔是 T（见图 4-21）。面积的变化是 $\frac{1}{2}T(x[k]+x[k-1])$。因此，如果输出是以前面积与面积的变化之和，则差分方程是

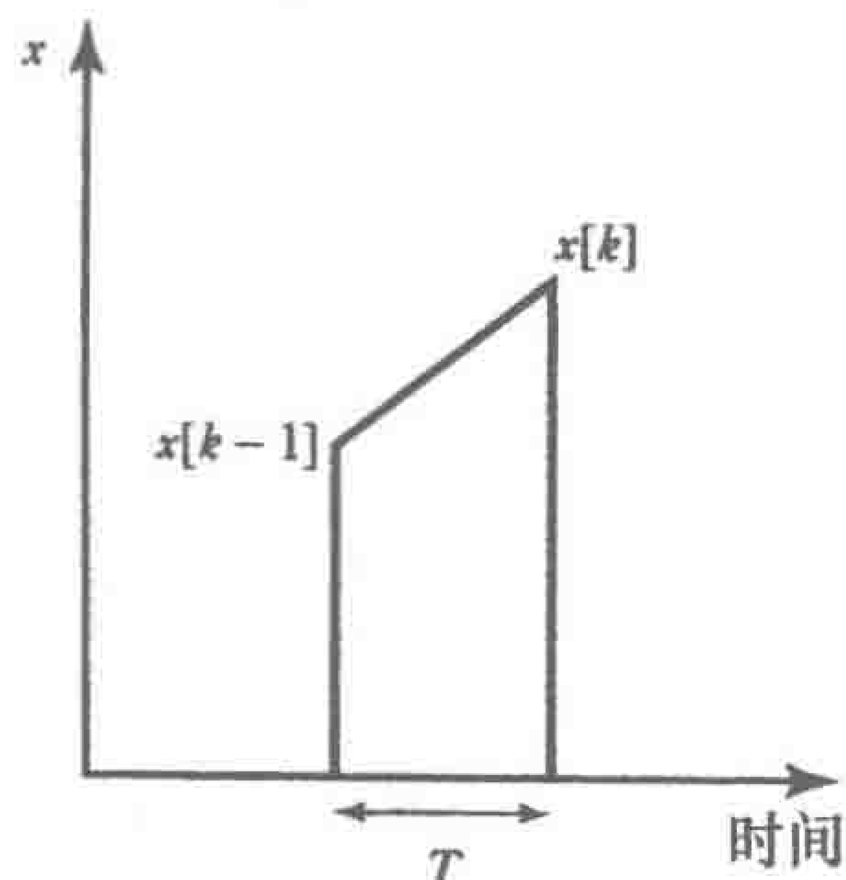


图 4-21　积分

$$y[k] = y[k-1] + \frac{1}{2}T(x[k] + x[k-1])$$

这称为积分的塔斯廷逼近（Tustin's approximation for integration）。

微分运算可以近似确定输入的变化率。因此，在时间 T 内，当输入从 $x[k-1]$ 变化到 $x[k]$时，输出是

$$y[k] = (x[k] - x[k-1])/T$$

本章小结

模拟数字转换是把模拟信号转换成二进制字。时钟提供一个周期性的时间信号给 ADC，在每一个时钟脉冲出现时对模拟信号进行采样。采样和保持单元保持每个采样值，直到出现下一个时钟脉冲。ADC 的形式有逐次逼近式 ADC、斜坡式 ADC、双斜积分式 ADC 和快闪式 ADC。

数字模拟转换是把二进制字转换成模拟信号。DAC 的形式有加权电阻和 R-2R 梯形。

多路复用器是一个能够获得多个数据输入来源的电路，它选择其中一个输入通道的数据作为输出。

数据采集（或者 DAQ），用于接收来自传感器的数据，并把这些数据输入到计算机中进行处理。

数字信号处理或**离散时间信号处理**是微处理器用于处理信号的两个术语。

习题

4.1 模拟信号输入范围为 100V，字长 12 位的 ADC 的分辨率是多少？

4.2 传感器提供 5V 的最大模拟输出。如果分辨率是 10mV，ADC 的字长是多少？

4.3 图 4-6 中 R-2R DAC 的梯形电阻，其输出通过反相运算放大器的反馈电阻 2R 反馈。如果这个 DAC 是 3 位的且参考电压为 5V，确定转换器的分辨率。 110

4.4 对于一个二进制加权电阻 DAC，加权的 4 位 DAC 的输入电阻分别是多少？

4.5 时钟频率是 1MHz 的 12 位 ADC 的转换时间是多少？

4.6 在监测多个热电偶的输入时，下列模块的顺序用于每个热电偶与微处理器接口：
保护、冷端补偿、放大器、线性化、采样和保持器、模拟数字转换器、缓冲器、多路复用器。
解释每个模块的功能。

4.7 推荐可能的模块来连接微处理器与驱动器输出。

4.8 图 4-5 所示的 4 位加权电阻 DAC，确定输入分别为 0001、0010、0100 和 1000 的输出放大器的输出，输入 0V 为逻辑 0，5V 为逻辑 1。

4.9 如果在一个 16 位的加权电阻 DAC 中最小电阻为 R，那么最大电阻应该是多少？

4.10 10 位的斜坡式 ADC 的满量程输入是 10V。如果时钟周期为 15μs，那么需要多长时间转换这样一个满量程输入？

4.11 对于 12 位 ADC 满量程输入，逐次逼近式 ADC 比斜坡式 ADC 快多少？ 111

第5章 数字逻辑

本章目的

通过本章的学习，读者将能够：

- 识别逻辑AND门、OR门、NOT门、NAND门、NOR门和XOR门的逻辑符号，掌握实际应用，了解各种逻辑门系列电路的意义。
- 说明如何在控制系统中使用SR触发器、JK触发器和D触发器。
- 说明译码器和555定时器的操作方法。

5.1 数字逻辑简介

许多控制系统在满足一定条件时才开始或停止运动。例如，对于家用洗衣机，仅当滚筒内有水且达到指定水位时，才会启动加热器。这种控制只涉及两种可能电平的数字信号。数字电路是数字计算机和微处理器控制系统的基础。

我们可以把**数字控制**应用于家用洗衣机上。例如，如果关闭洗衣机门和洗衣机完成一个特定的工作循环，洗衣机才开始注水。有两个为yes或no的输入信号和一个为yes或no的输出信号。如果两个输入信号全为yes信号，则控制器输出yes信号，换言之，当输入信号A和B全为1时，输出变量为1。这种操作是由**逻辑门**控制的，上述的例子是AND（与）门。许多机器和处理过程都用这种方法来进行控制。**组合逻辑**用于组合两个或多个基本逻辑门形成所需的功能。例如，如果车钥匙插入点火开关的同时车门没关闭，或者车灯亮着的同时车门没关闭，蜂鸣器就会发出声音。组合逻辑只取决于某一特定瞬间的输入值。

除了讨论组合逻辑外，本章还讨论**时序逻辑**。这种数字电路用于执行特定序列的控制，并由时钟或“启用/禁用”控制信号所决定。这些是带存储器的组合逻辑电路。因此，输入信号的时间或历史序列也会影响输出信号。

112

5.2 逻辑门

逻辑门是数字电子电路的基本组成模块。

5.2.1 AND门

假设有一个门电路仅当输入A和输入B都为高电平时，输出才为高电平；对于所有其他情况，输出为低电平。这样的门称为AND（与）逻辑门。可以把AND门想象为两个开关串联所组成的电路（见图5-1a）。只有当A与B同时接通时才有电流通过。不同的逻辑门使用不同的标准电路符号，其主要形式起源于美国。然而，现在已经制定出了国际标准形式（IEEE/ANSI）。它去除了特殊的形状，并在矩形框里填入逻辑函数。图5-1b表示AND门的美国符号形状，图5-1c表示新标准符号，&符号表明AND逻辑。两种形式都会在本教材中使用。如图5-1所示，可以用方程来表达AND门的输入和输出之间的关系，称为**布尔方程**（见附录C）。AND门的布尔方程为：

$$A \cdot B = Q$$

AND门的一个例子是应用于机器工具上的连锁控制系统，如果安全防护装置是得当的并给出1信号，同时电源打开，给出1信号，则输出1信号，机器开始工作。另一个例子是防盗报警器，当门打开时传感器被激活，并给出一个输出信号，报警器就会开启并响起来。

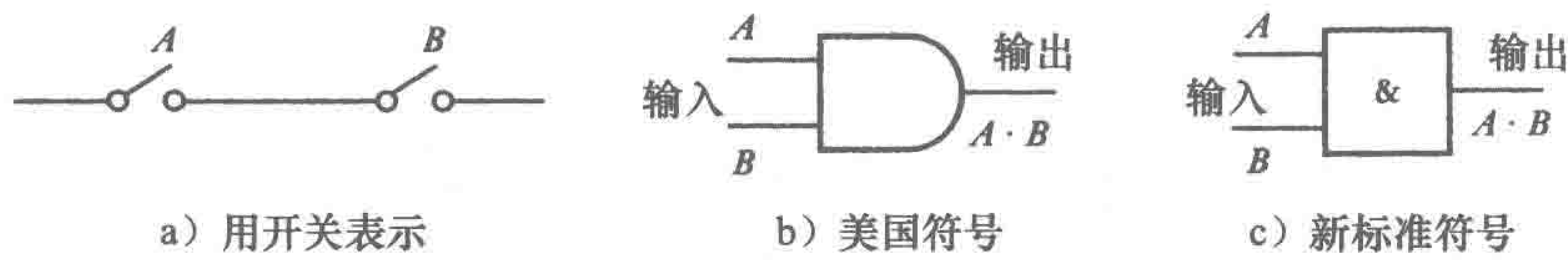


图5-1 AND门

逻辑门的输入与输出之间的关系可制成一个列表，称为**真值表**。真值表指定了输入与输出的关系。因此，对于具有输入 A 与 B 以及一个输出 Q 的AND门，当且仅当 $A=1$ 和 $B=1$ 时，输出才为1。A 和 B 的所有其他组合会输出0。AND门的真值表为：

输入		输出
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

113

如图5-2所示，思考当两个数字输入是时间的函数时会发生什么。该图称为AND门的时序图。仅当各输入为高电平时，AND门的输出才为高电平，该输出结果如图5-2所示。

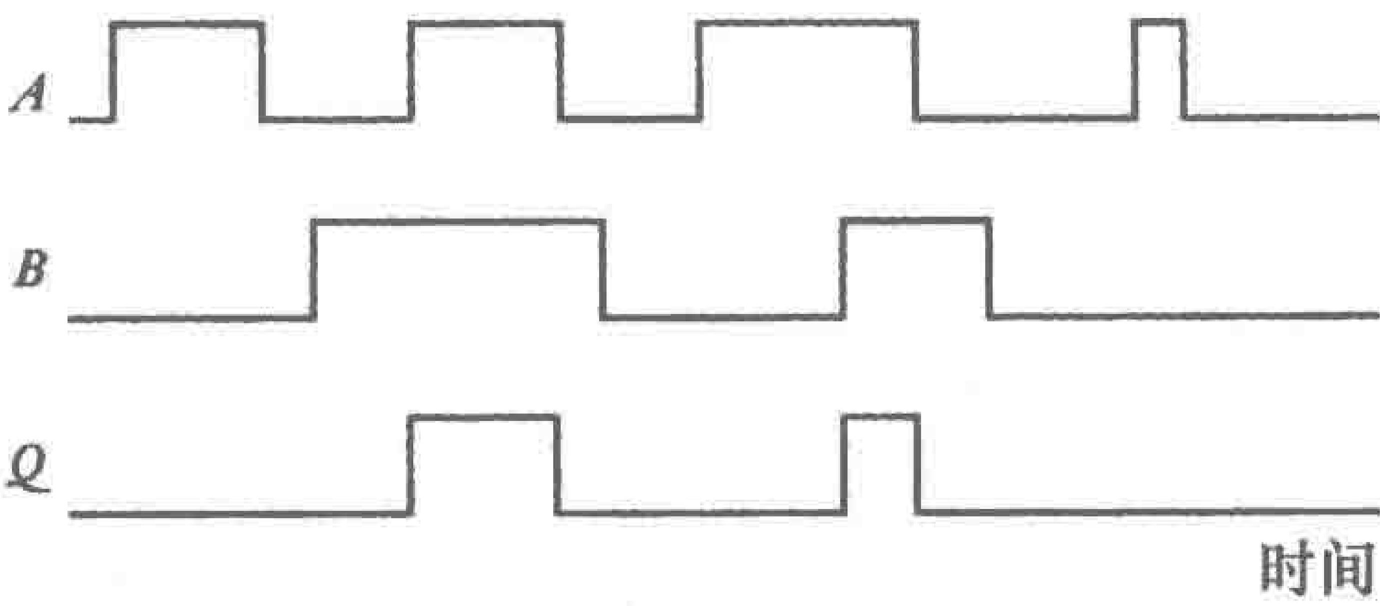
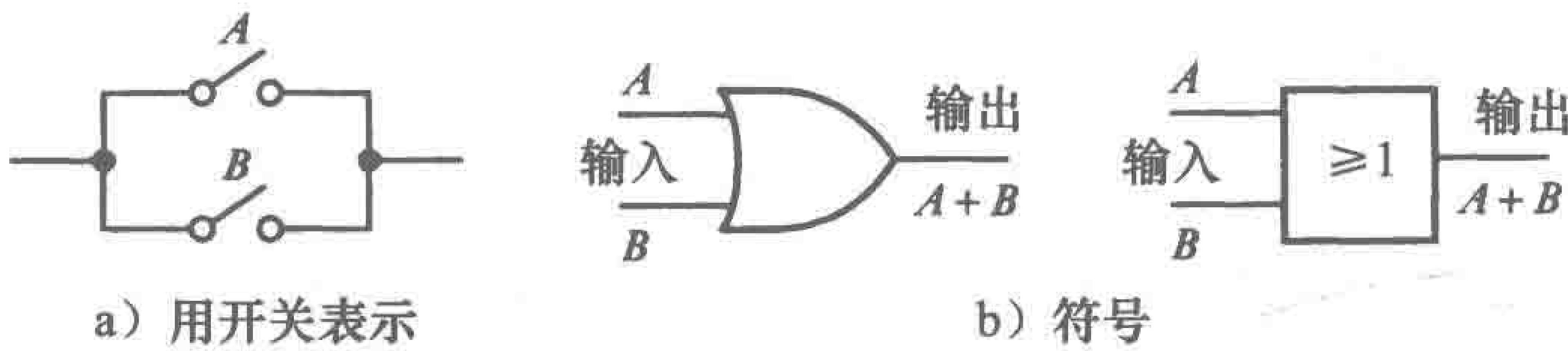


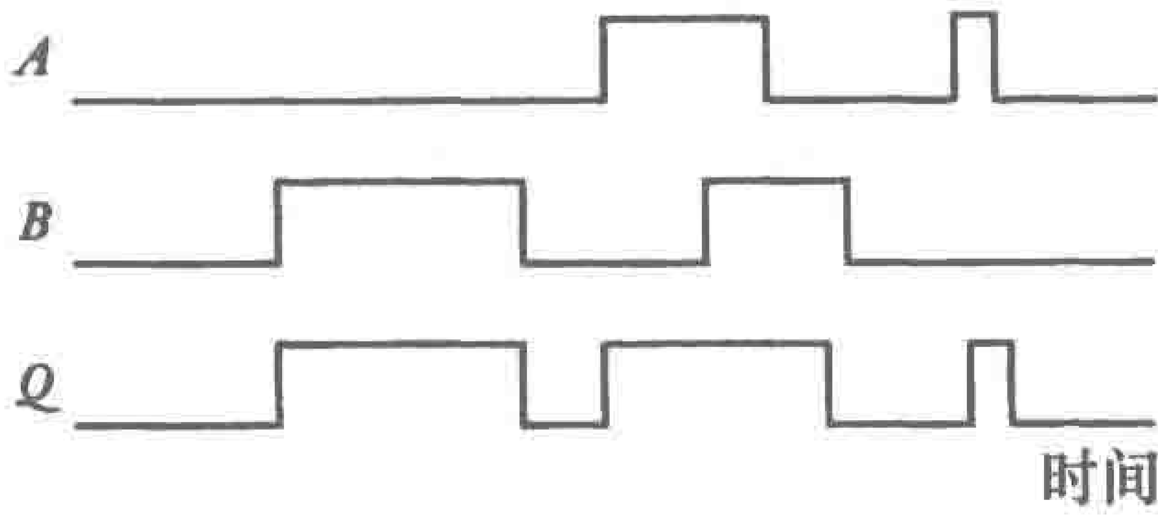
图5-2 AND门的时序器

5.2.2 OR门

当OR（或）门的两个输入 A 与 B 中至少有一个为1时，输出为1。可以把OR门想象成两个开关并联所组成的电路（见图5-3a）。当开关 A 或 B 闭合时，就会有电流通过。OR门可以有两个以上的输入。OR门的真值表为：

输入		输出
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1





c）时序图

图5-3 OR门

OR门的布尔方程为

$$A+B=Q$$

114

OR门的逻辑符号如图5-3b所示，方框内使用一个大于等于1的符号来描述OR门。如果至

少有一个输入为真，则 OR 函数为真。图 5-3c 为 OR 门的时序图。

5.2.3　NOT 门

NOT（非）门只有一个输入和一个输出。当输入为 0 时，输出为 1；当输入为 1 时，输出为 0。NOT 门的输出和输入是相反的，称为**反相器**（见图 5-4a）。方框中的 1 代表逻辑非的逻辑符号，即空操作，其输出端的圆圈代表取反的意思。因此，如果数字输入随时间的变化如图 5-4b 所示，则输出随时间的变化是反相的。

图 5-4　NOT 门

NOT 门的真值表如下：

输入 A	输出 B
0	1
1	0

NOT 门的布尔方程为

$$\overline{A} = Q$$

符号上的横线表示反相，或求反，因此 A 上的横线表示输出 Q 是 A 的反相值。

5.2.4　NAND 门

NAND（与非）门可以理解为 AND 门和后面的 NOT 门的组合（见图 5-5a）。因此，当输入 A 与输入 B 都为 1 时，输出为 0；对于所有其他输入，则输出 1。

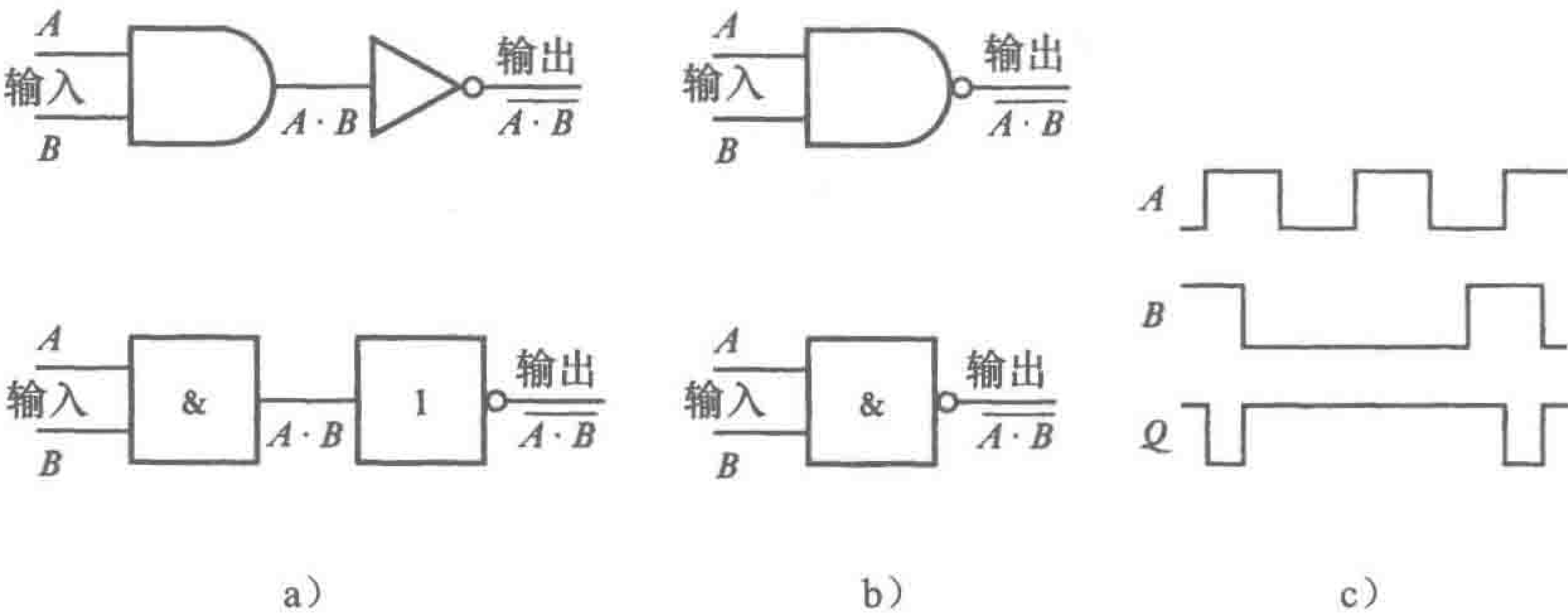


图 5-5　DAND 门

NAND 门只是把 AND 门真值表的输出做了反相处理。考虑这个门的另一种方式是先对两个输入信号进行 AND 运算，再对运算结果进行 NOT 运算得到。图 5-5b 为 NAND 门的逻辑符号，紧随逻辑与符号后的圆圈表示进行反相处理。NAND 门的真值表如下：

115

输入		输出
A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

NAND 门的布尔方程为

$$\overline{A\cdot B} = Q$$

图 5-5c 说明了当两个输入都为随时间变化的数字信号时 NAND 门的输出结果。只有两个

输入都为高电平时，输出才为低电平。

5.2.5 NOR门

NOR（或非）门可以理解为OR门和后面的NOT门的组合（见图5-6a）。因此，当输入A和输入B至少有一个为1时，输出为0。NOR门只是把OR门的输出做了反相处理。考虑这个门的另一种方式是先对两个输入信号进行OR运算，再对运算结果进行NOT运算得到。图5-6b为NOR门的逻辑符号，紧随逻辑或符号后的圆圈表示进行反相处理。NOR门的布尔方程为

$$\overline{A+B}=Q$$

图5-6c为NOR门的时序图，其真值表如下：

输入		输出
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

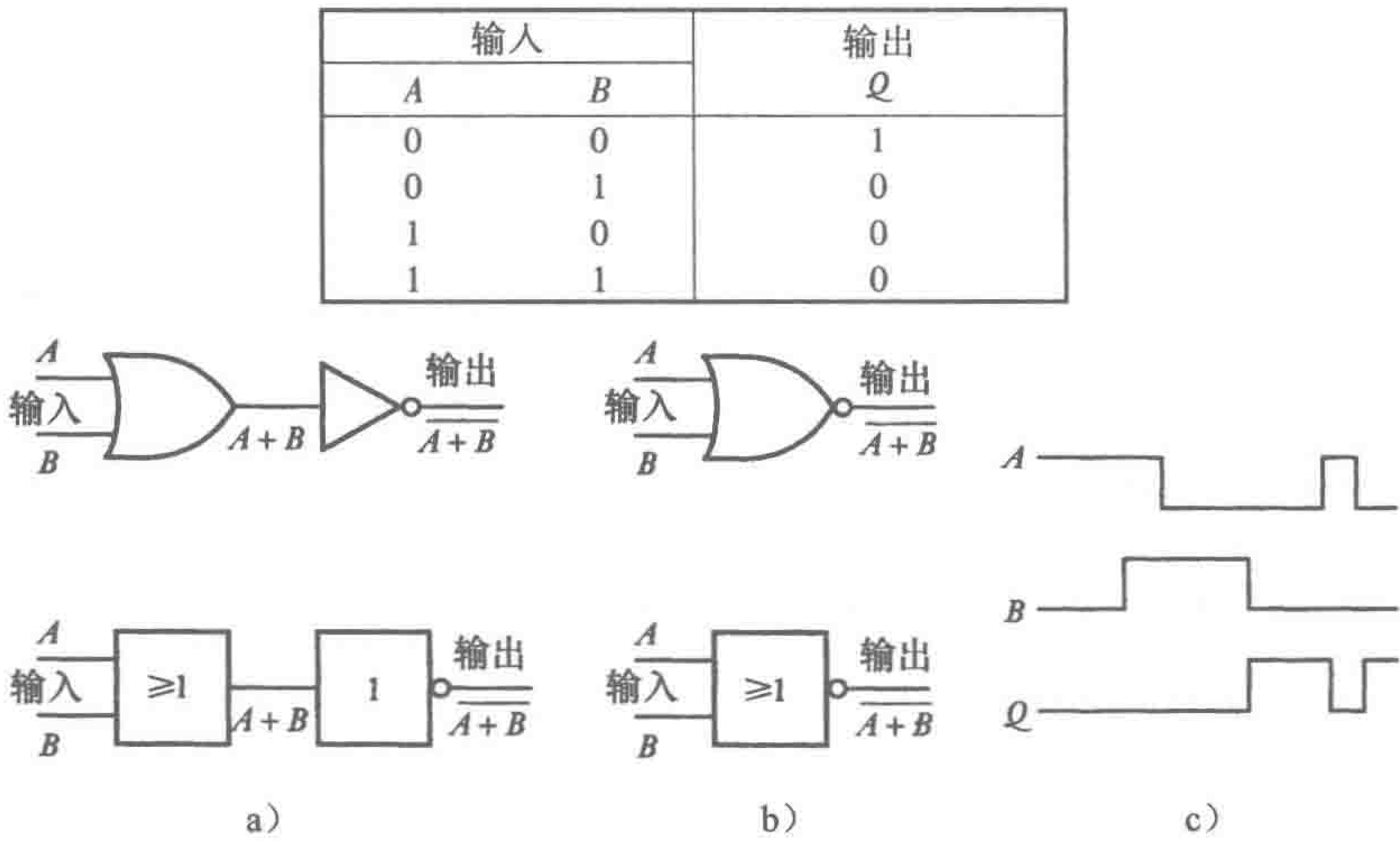


图5-6 NOR门

5.2.6 XOR门

XOR（异或）门可以理解为先对两个输入信号A和B进行OR运算，同时对这两个输入信号进行NAND运算，再对上面两个运算的结果进行AND运算。若两个输入A、B的电平相异，则输出为高电平1；若两个输入的电平相同，则输出为低电平0。XOR门的逻辑符号如图5-7b所示，=1表示仅当一个输入为真时，输出为真。图5-7c为XOR门的时序图，其真值表如下：

输入		输出
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

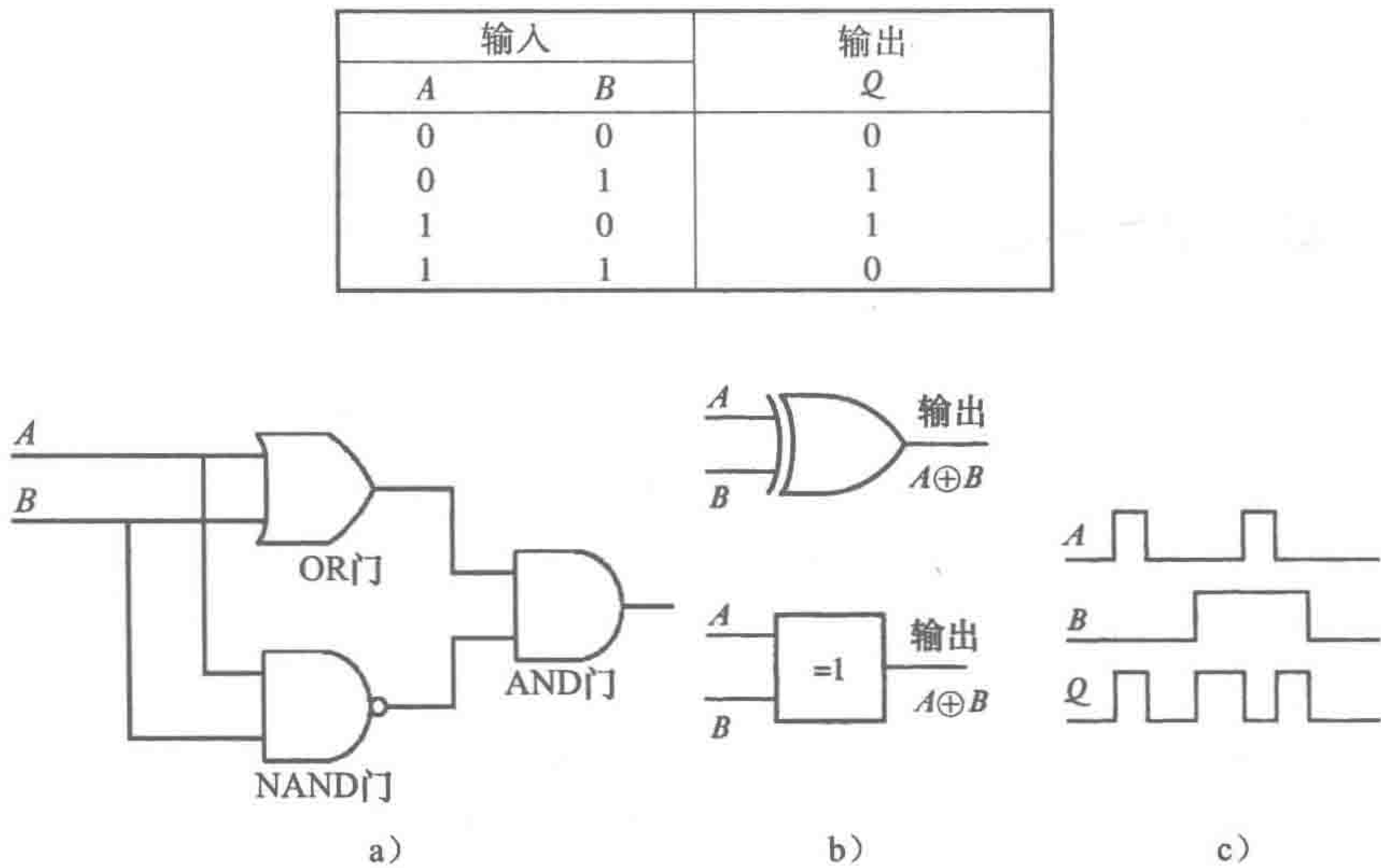


图5-7 XOR门

5.2.7　逻辑门的组合

116 ~ 117

为了得到逻辑系统，需要一些门来完成。然而，如图 5-8 所示，可以将所有的门组成一个门。考虑图 5-8 中 3 个 NOR 门的组合。

包含中间输出和最终输出的真值表如下所示：

A	B	C	D	Q
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

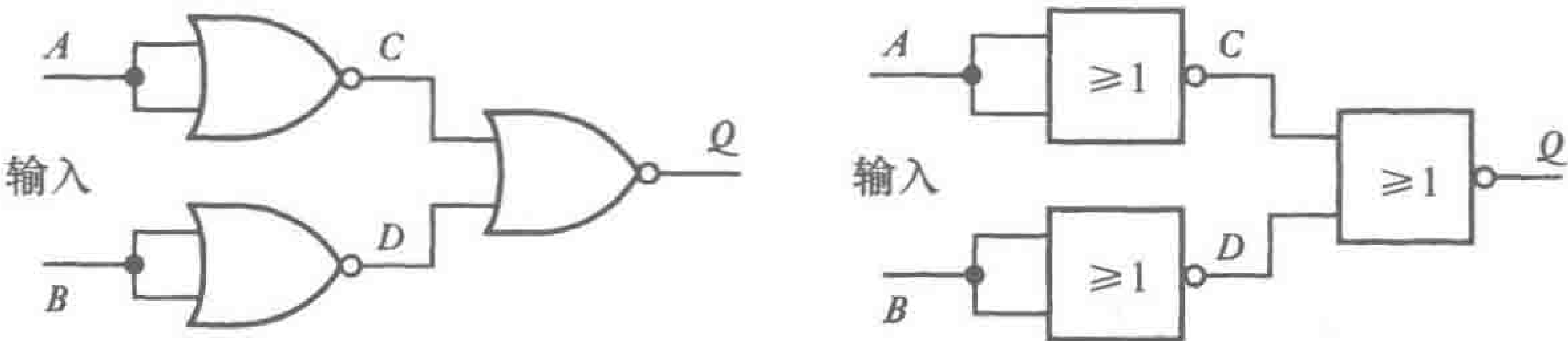


图 5-8　3 个 NOR 门

它的结果和 AND 门的结果相同。如果在这样的组合后加一个 NOT 门，则其真值表和 NAND 门相同。

图 5-9 为 3 个 NAND 门的组合，其真值表包含中间输出和最终输出，如下所示：

A	B	C	D	Q
0	0	1	1	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	1

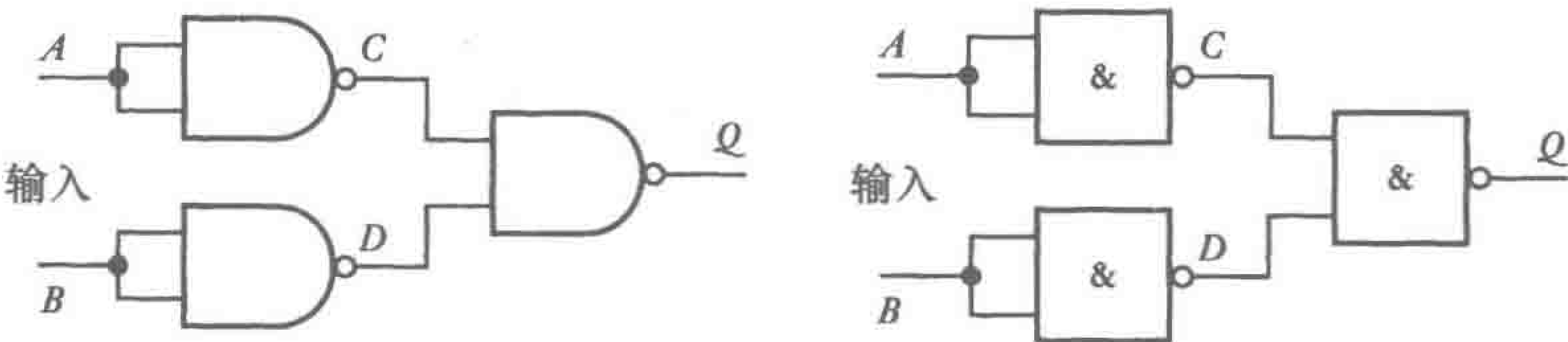


图 5-9　3 个 NAND 门

它的结果和 OR 门的结果相同。如果在这样的组合后加一个 NOT 门，则其真值表和 NOR 门相同。

如果使用以上多个门，则上述两种门的组合表明如何用一种类型的门（例如，NOR 门或者 NAND 门）来代替其他门。逻辑门也可以用来组成复杂逻辑门电路和时序电路。

逻辑门可以通过集成电路得到。不同的制造商已经规范了编码方法，使得无论哪个制造商，基本零件编号都是相同的。例如，图 5-10a 所示的逻辑门系统可用 7408 集成电路得到，它有 4 个双输入的与门，采用 14 针双列直插封装。电源连接到引脚 7 和引脚 14，给 4 个与门供电。为了从外部封装能够看出引脚 1 在哪一头，在引脚 1 和引脚 14 之间做了一个缺口。7411 集成电路有 3 个与门，每个门有 3 个输入；7421 集成电路有两个与门，每个门有 4 个输入。图 5-10b 所示的逻辑门系统可在 7402 集成电路中得到。它有 4 个双输入或非门，采用 14 针双列直插封装，电源连接到引脚 7 和引脚 14。7427 集成电路有 3 个门，每个门有 3 个输入；7425

118

集成电路有两个门，每个门有4个输入。

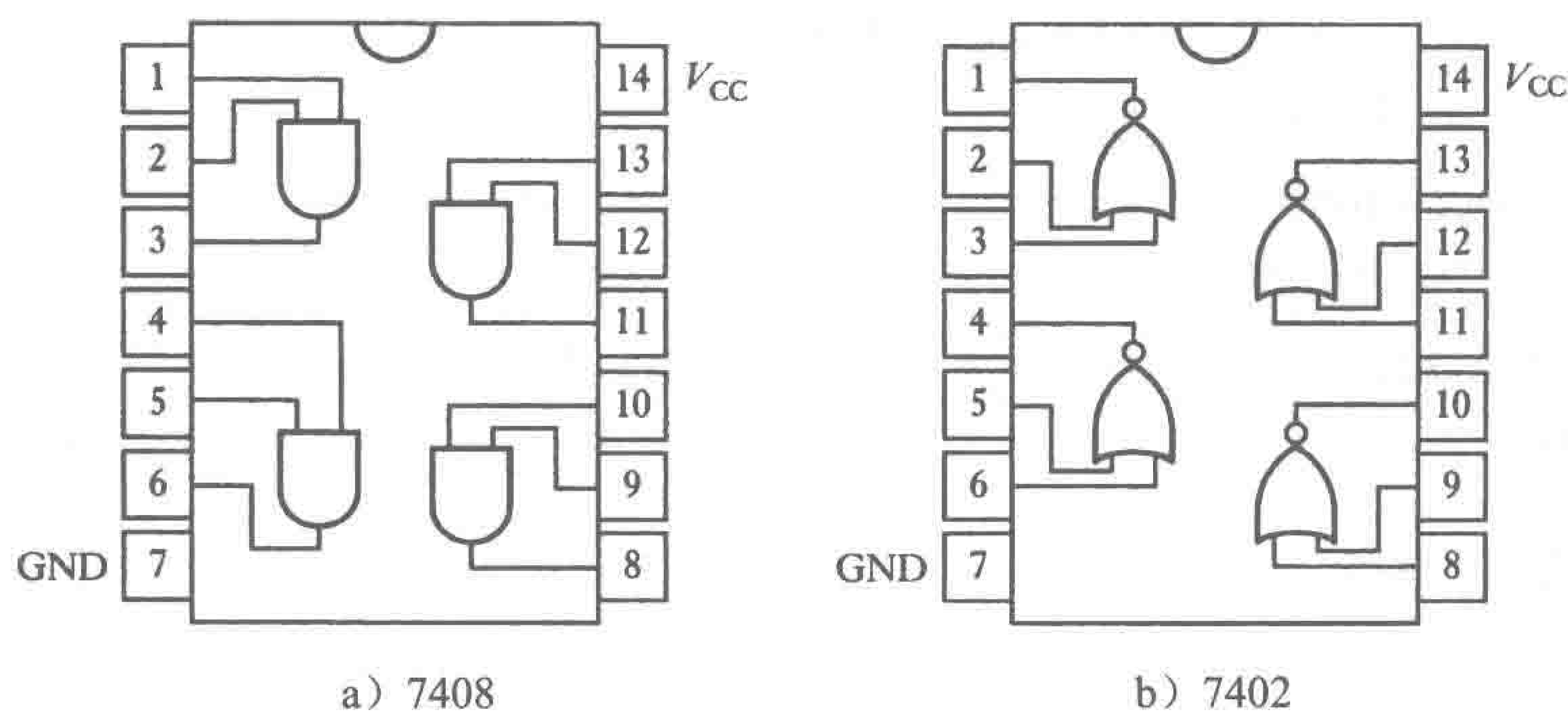


图5-10 集成电路

对于如何使用布尔代数和德·摩根（De Morgan）定律、卡诺图（Karnaugh map）等技术从多个逻辑门生成需要的逻辑功能的讨论，请参照附录C。

5.2.8 逻辑系列与集成电路

为了实现数字逻辑设计，我们有必要理解逻辑系列的重要性以及它们不同的操作原则。由相同的技术和电特性构成的集成电路组成一个**逻辑系列**（logic family）。常见的逻辑系列有晶体管晶体管逻辑（TTL）、互补金属氧化物半导体（CMOS）和射极耦合逻辑（ECL）。一般的参数概述如下。

1）**逻辑电平**，即能够对应于二进制1和0状态的电压电平范围。对于标准74XX TTL系列，典型的对应于二进制0状态的电压范围是0～0.4V，而对1状态而言，则是2.4～5.0V之间。对于CMOS，其电压电平取决于使用的电源电压V_{DD}。它们的范围为3～15V，逻辑1的最小电压为$0.7V_{DD}$时，逻辑0的最大电压为$0.3V_{DD}$。

2）**抗干扰度或噪声容限**，即在输出电压没有发生失真变化时电路对噪声的抗干扰能力。对于标准74XX TTL系列，噪声容限是0.4V。因此，对于输入逻辑0和逻辑1并且要求它们仍然被看作0和1时，可以接受的误差是0.4V。对于CMOS，噪声容限取决于电源电压并且为$0.3V_{DD}$。

3）**扇出系数**，指的是当一个标准门输出保持所需要的电平状态（高或低电平）时，它所能驱动的同类门的个数。这取决于一个门能提供多少电流和驱动一个逻辑门需要多少电流。对于标准TTL门，扇出系数为10；对于CMOS，扇出系数是50；对于ECL，扇出系数是25。如果更多的门连接到主动门上，那么将无法提供足够的电流来驱动它们。

119

4）**拉电流或灌电流行为**，即电流如何在一个逻辑门的输出端和另一个门的输入端之间流动。例如一个门驱动另一个门的情况，“拉电流”是指高电平时驱动门对下一个门的输入端供应电流。“灌电流”是指当驱动门低电平时接收从从动门流回的电流。TTL门是以“灌电流”的形式运行的。

5）**传播延迟时间**，即数字电路对输入电平变化响应的快慢。通常TTL门的延迟时间为2～40ns，一般比CMOS门快5～10倍，但比ECL门慢（其传播延迟通常为2ns）。

6）**功率损耗**，即逻辑门从电源消耗的功率。TTL每个门约消耗10mW的功率，而CMOS除了切换开关外，均不消耗能量。ECL每个门大约消耗25～60mW的功率。

决定应该使用哪些逻辑系列的主要准则是传播延迟和功率损耗。与TTL相比，CMOS的优势是功率低，这使它适合于一些用干电池供电的装置。来自不同逻辑系列的集成电路连接在一起也是可行的，但这需要应用一些特殊的接口技术。

TTL系列的应用很广泛，该系列常称为74XX系列。该系列有不同的形式。通常，标准TTL是7400，其功率损耗为10mW，传播延迟为10ns。低功耗的肖特基TTL(LS)是74LS00，

其功率损耗为2mW，传播延迟与上面的相同。先进的低功耗肖特基TTL(ALS)是74ALS00，其更快且功率损耗更少，传播延迟为4ns，功率损耗为1mV。快速TTL(F)是74F00，其传播延迟为3ns，功率损耗为6mW。

CMOS系列（包括4000系列）虽然比TTL系列的功率损耗低，但速度却要远远低于TTL系列。40H00系列更快些，但仍然比TTL(LS)慢。74C00系列与TTL系列引脚兼容，使用相同的编码系统，但名称从74C开始。虽然74C00比TTL系列有功耗上的优势，但速度上还是低于TTL系列。74HC00和74HCT00与TTL(LS)系列相比速度更快。

5.3　逻辑门的应用

以下是一些简单的应用逻辑门的实例。

5.3.1　校验位发生器

第4章讨论了使用校验位来检测错误的方法。在每个代码块中增加一位，强制使包括校验位在内的位数成为奇数（如果应用奇数检验）或偶数（如果应用偶数检验）。

图5-11的逻辑门电路可用于判断和添加适当的校验位。该系统采用XOR门，如果所有输入为0或1，则该门输出0；如果输入不同，则输出1。进行双位检查时，如果它们不同，则输出1。如果使用奇数校验，偏置位就是0；如果使用偶数校验，则是1。合适的偏置位可以添加到该信号进行传输。同样的电路也可在接收端进行奇偶检测，有错误则最终输出为1。这样的电路可通过集成电路的形式得到。

120

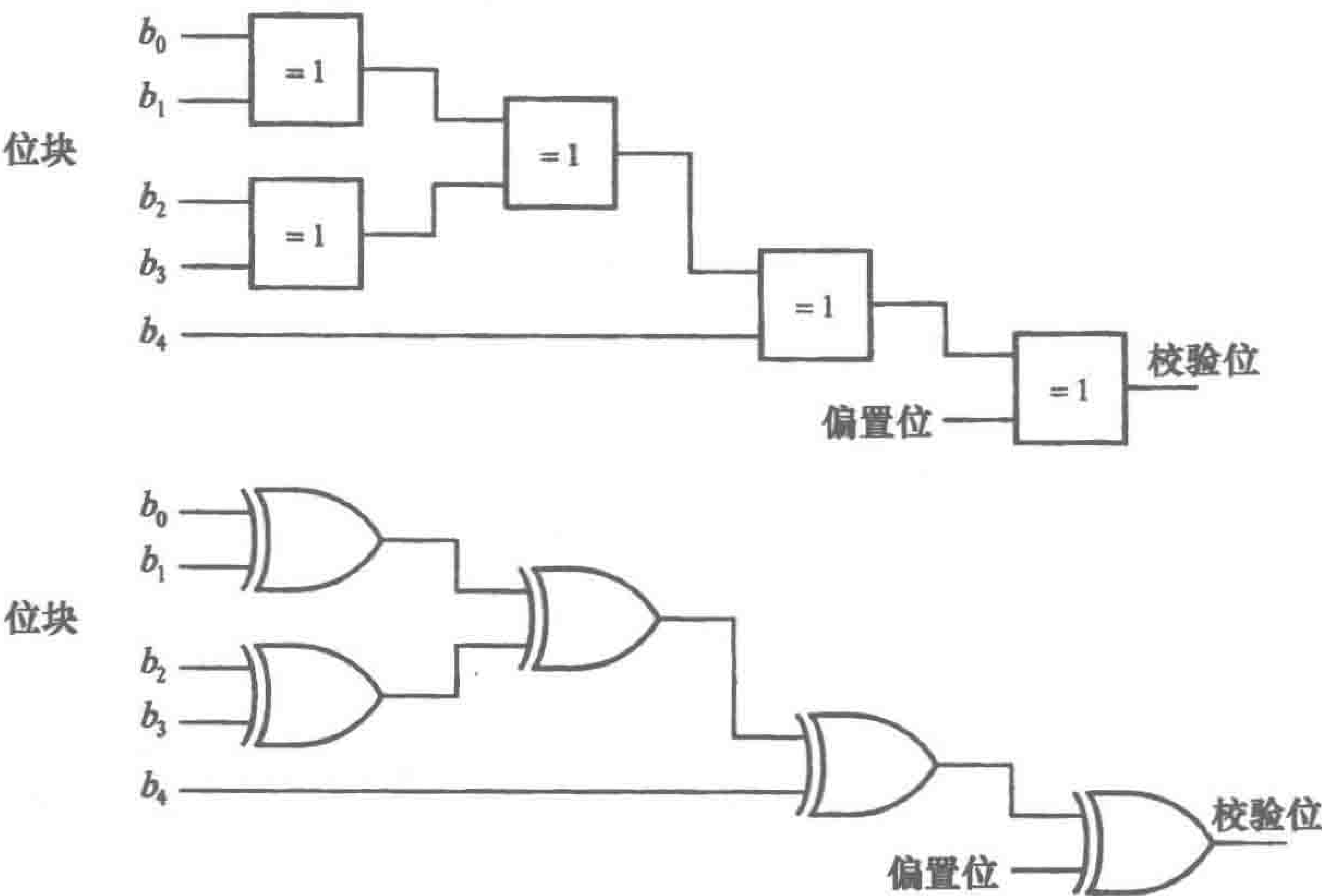


图5-11　校验位发生器

5.3.2　数字比较器

数字比较器用于比较两个数字字，判断它们是否完全相等。两个字按位进行比较，如果相同，则输出1。XOR门可用来比较两个位是否相等。如果两个位均为0或1，则输出0；而当它们不相等时，则输出1。如果希望位相同时输出1，则需要添加NOT门，这种XOR门和NOT门的组合称为XNOR（异或非）门。为了比较两个字中的每一对“位”，每对“位”都需要一个XNOR门。如果每对都相同，则每个XNOR门输出1。当所有XNOR门输出1时，可用一个AND门输出1。图5-12显示了此系统。

数字比较器可应用在集成电路中，不仅可以判断两个字是否相同，还可判断两个字的大

小。例如，7485四位比较器对两个四位的字A和B进行比较，如果A大于B，则引脚5输出1；如果A等于B，则引脚6输出1；如果A小于B，则引脚7输出1。

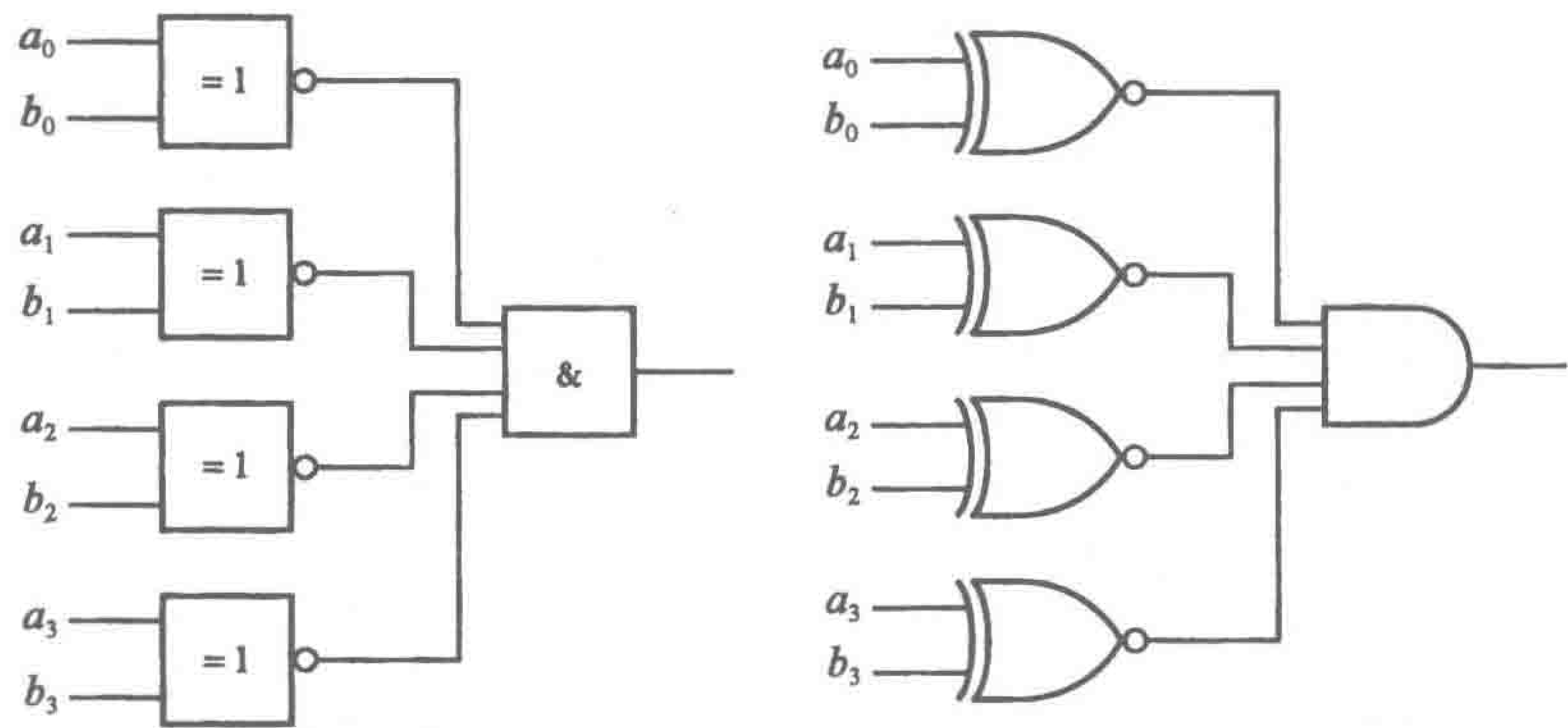


图5-12 比较器

5.3.3 编码器

图5-13是一个通过控制器发送数字编码信号到交通信号灯上的简单系统，用代码决定点亮红、黄或绿灯。为了点亮红灯，传输信号$A=0$，$B=0$；为了点亮黄灯，传输信号$A=0$，$B=1$；为了点亮绿灯，传输信号$A=1$，$B=0$。可以通过三个AND门和两个NOT门，使用三种编码来点亮信号灯。

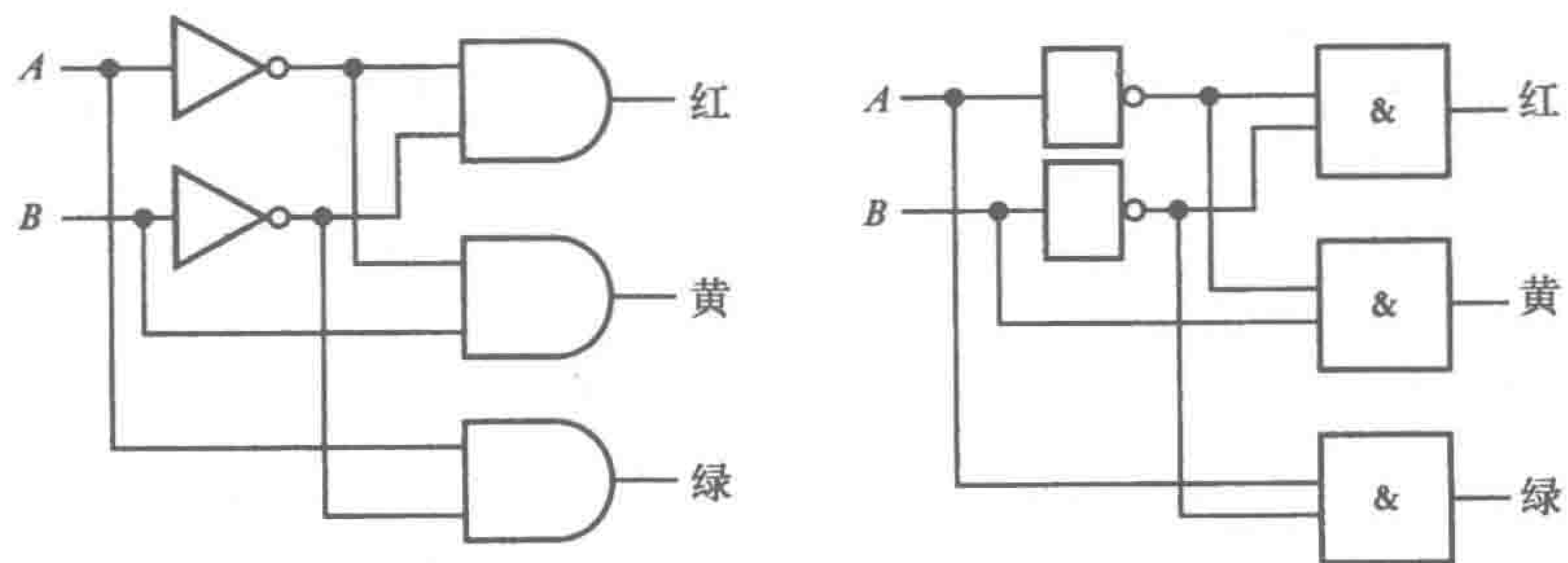


图5-13 交通信号灯

5.3.4 译码器

在许多应用中需要把一种代码转换成另一种代码。例如，微处理器系统的输出为BCD（二－十进制代码，Binary-Coded Decimal），需要转换成合适的代码来驱动七段数码显示器。**数据译码**是指将某一组编码（如BCD、二进制、十六进制）转换成某一个代表这一组编码的有效输出。对于n位字的编码输入来说，译码器有n位二进制输入线，对应地给出m位输出线，这样对于一个可能的输入组合只有一根线起作用，也就是说，对于一个特定字输入编码，只有一根输出线产生输出。例如，一个BCD－十进制译码器有4位输入编码和10根输出线，这样一个特定的BCD输入只会激活一根输出线。因此，用对应于每一个十进制数的输出线来表示某一特定的十进制数。

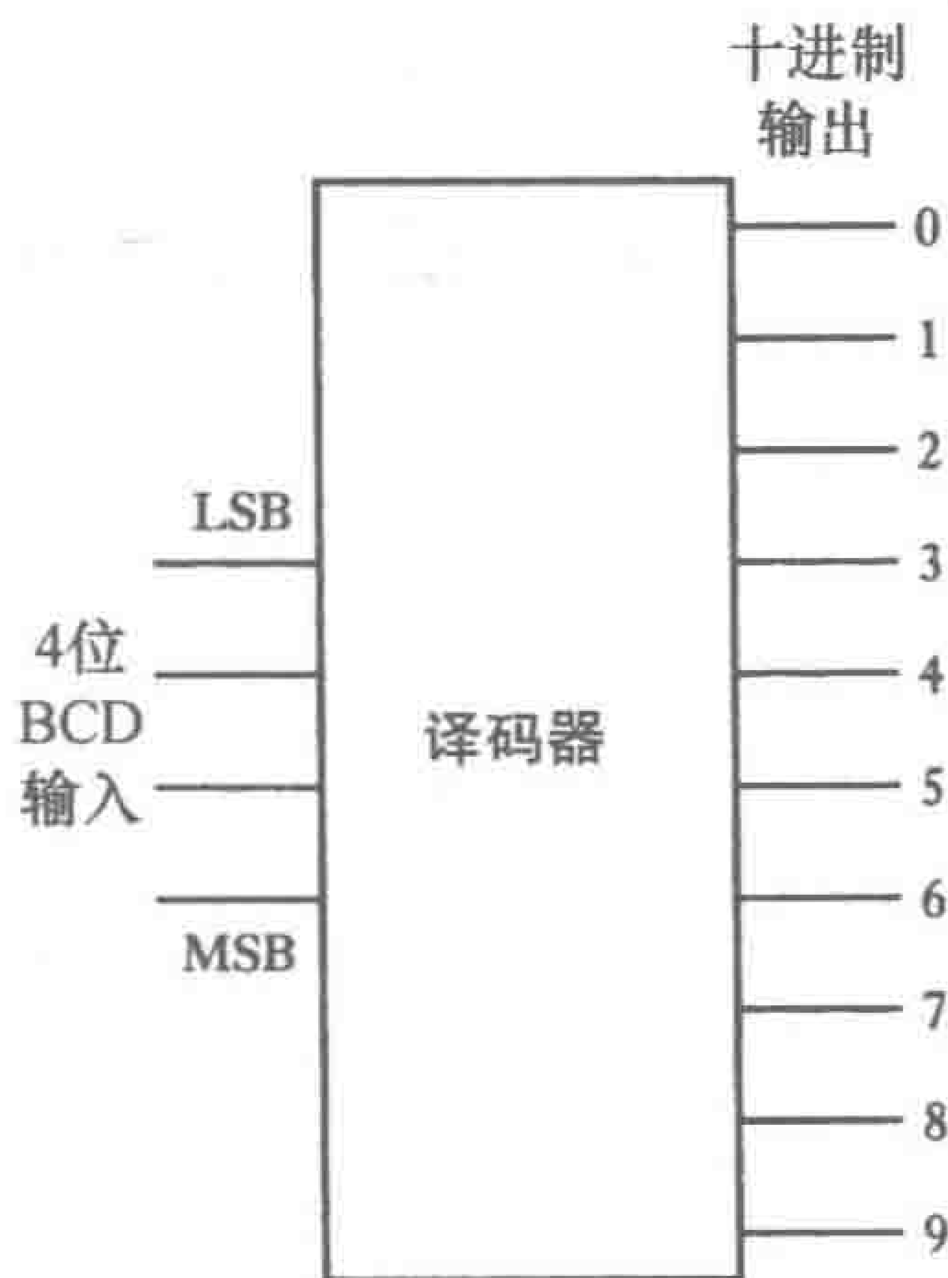
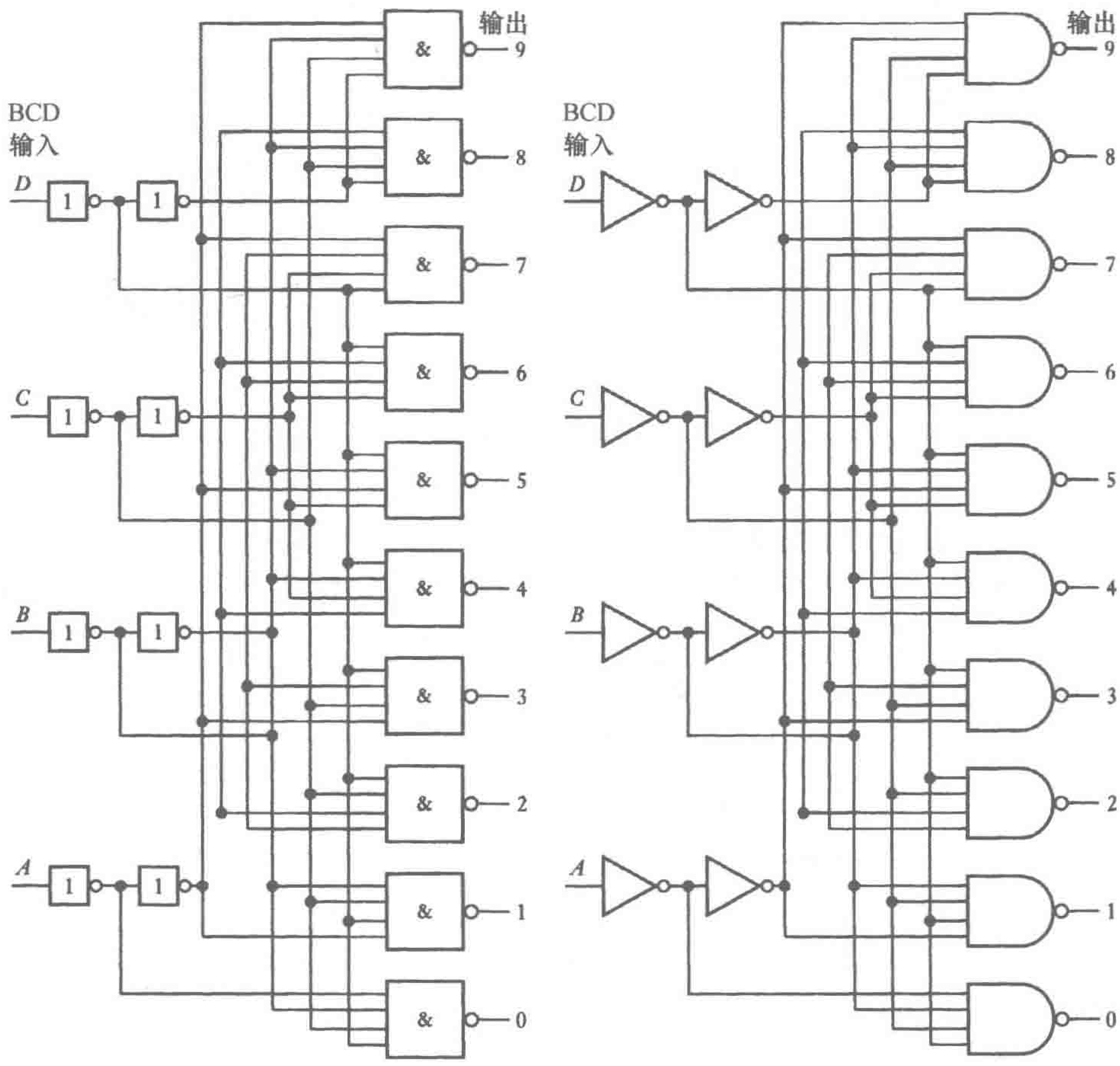


图5-14 译码器

总的来说，**译码器**是这样一种逻辑电路：它接受输入信号，判断其对应哪个数字，并激活与该数字相对应的输出端。译码器在微处理器电路中有广泛的应用。

121~122

译码器的输出端有两种形式：一种是高电平时表现为有效输出，低电平为无效输出；另一种正好相反。对于高电平有效输出，译码器可以用 AND 门进行设计；而对于低电平有效输出，可以用 NAND 门设计。图 5-15 显示的正是 BCD－十进制低电平有效输出译码器的内部电路图和真



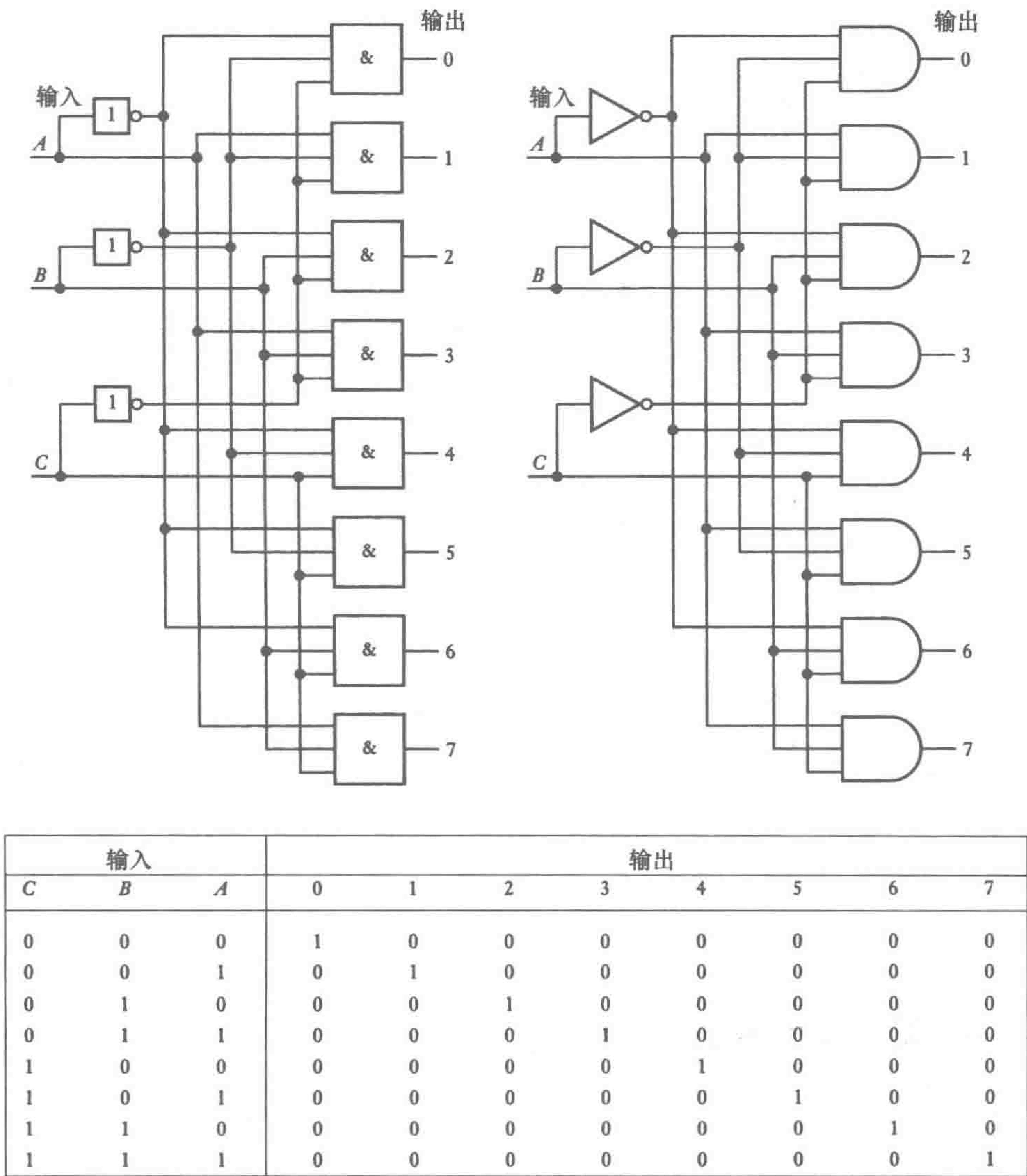
输入				输出									
A	*B*	*C*	*D*	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1	1
0	0	1	1	1	1	1	0	1	1	1	1	1	1
0	1	0	0	1	1	1	1	0	1	1	1	1	1
0	1	0	1	1	1	1	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

图 5-15　BCD－十进制译码器：1＝高电平，0＝低电平

值表。这样的译码器有现成的集成电路，例如 74LS145。

BCD－七段译码器也被广泛应用，其型号有 74LS247。它可以将输入的 4 位 BCD 信号转化成相应的信号驱动七段显示器。

3 线－8 线译码器有 3 个输入端、8 个输出端。它将输入的 3 位二进制数转化成对应该数的输出端，并激活它。图 5-16 展示了如何用逻辑门实现这样的译码器，以及其真值表。

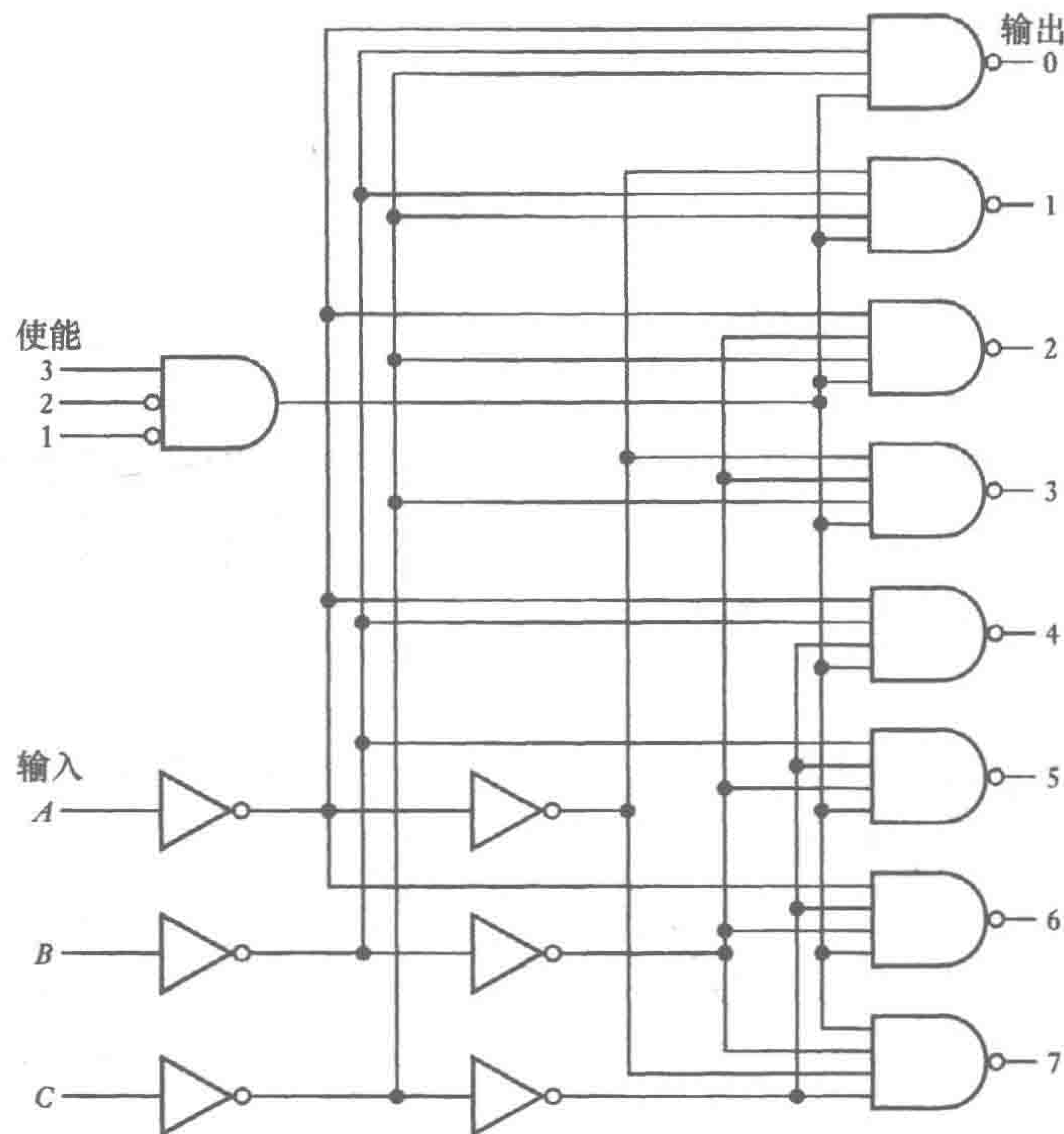


输入			输出							
C	*B*	*A*	0	1	2	3	4	5	6	7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

图 5-16　3 线－8 线译码器

有些译码器有一条或多条使能（ENABLE）线（用于控制译码器的运算）。如果使能端是高电平，译码器就能够正常工作；如果使能端是低电平，那么无论输入是什么，输出端都是低电平。图 5-17 展示的是常用的有使能端的 3 线－8 线译码器 74LS138。注意，图 5-17 中的译码器是低电平有效输出，而不是图 5-16 的高电平有效输出。此外，该译码器的使能端有 3 个端口，当 *E*1、*E*2 为低电平并且 *E*3 为高电平时，译码器正常工作。其他的变化值均禁止该译码

器，使得所有的输出线都是高电平。



使能			输入			输出							
*E*1	*E*2	*E*3	*C*	*B*	*A*	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

123
~
125

图 5-17 有使能端的 3 线 – 8 线译码器 74LS138：1 = 高电平，0 = 低电平，X = 无关

图 5-18 显示的是对于 74LS138 译码器，不同的输入得到的输出响应类型。

74LS138 译码器可和一个微处理一起使用，ENABLE 端打开译码器后，根据从微处理器的三根输出线的输出电平，将八位译码器的一位变为低电平，其他几位变为高电平。这样，如果每个输出器件都有一个地址，例如一个独一无二的二进制地址，那么微处理器将地址发送到译码器中，译码器就能够激活分配这个地址的器件。然后，74LS138 可用做一种地址译码器。

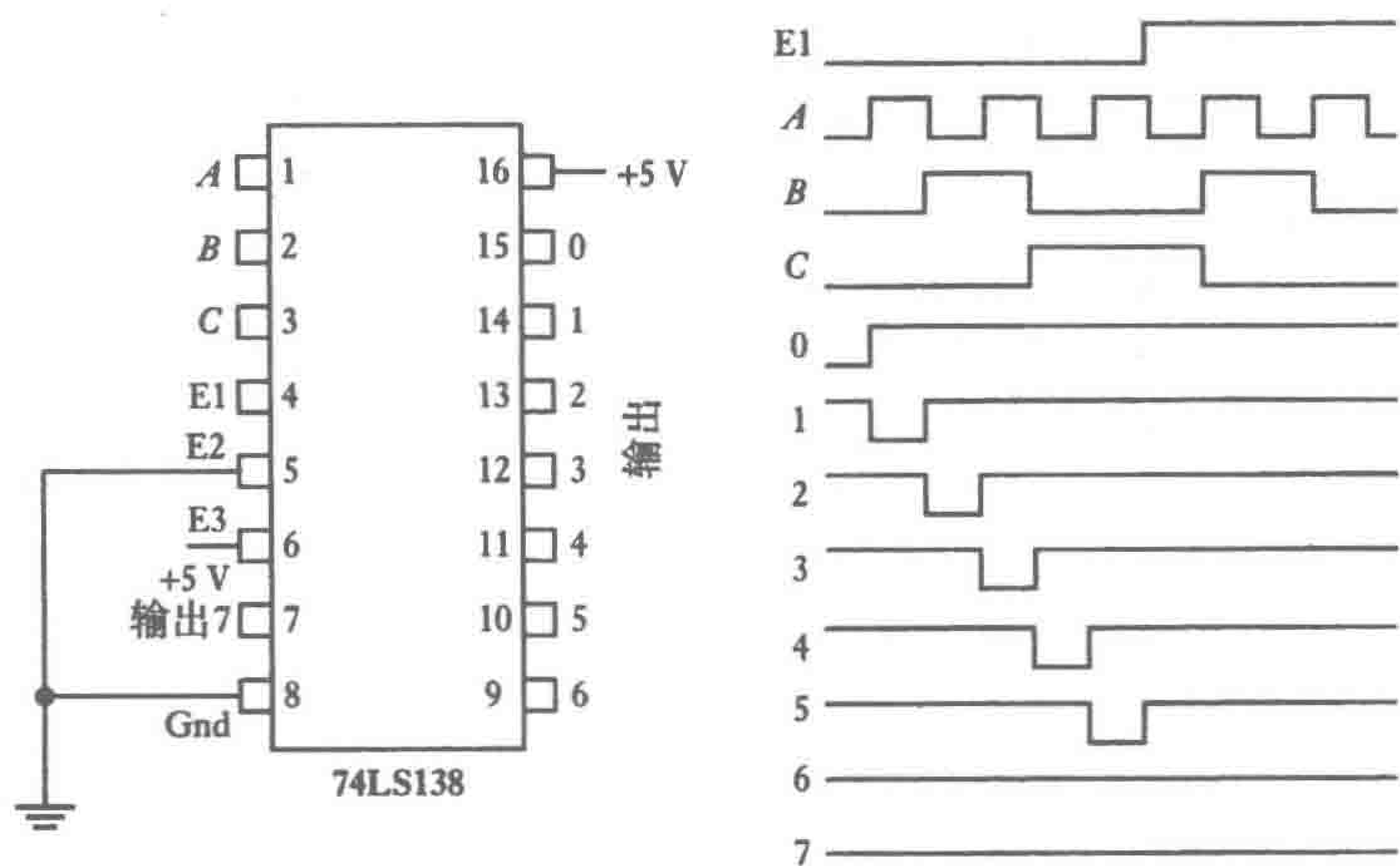


图 5-18 74LS138

5.4 时序逻辑

本章前面所述的逻辑电路都是组合逻辑系统。在这样的系统中，输出是由某一时刻输入变量的组合决定的。例如，如果输入信号 *A* 和 *B* 同时发生，那么 AND 门就会给出一个输出。输出信号并不依赖于之前输入信号是什么。当系统的输出需要依赖于之前的输入信号的值时，就需要**时序逻辑**系统。组合逻辑系统和时序逻辑系统的主要区别在于后者必须有某种形式的存储器。

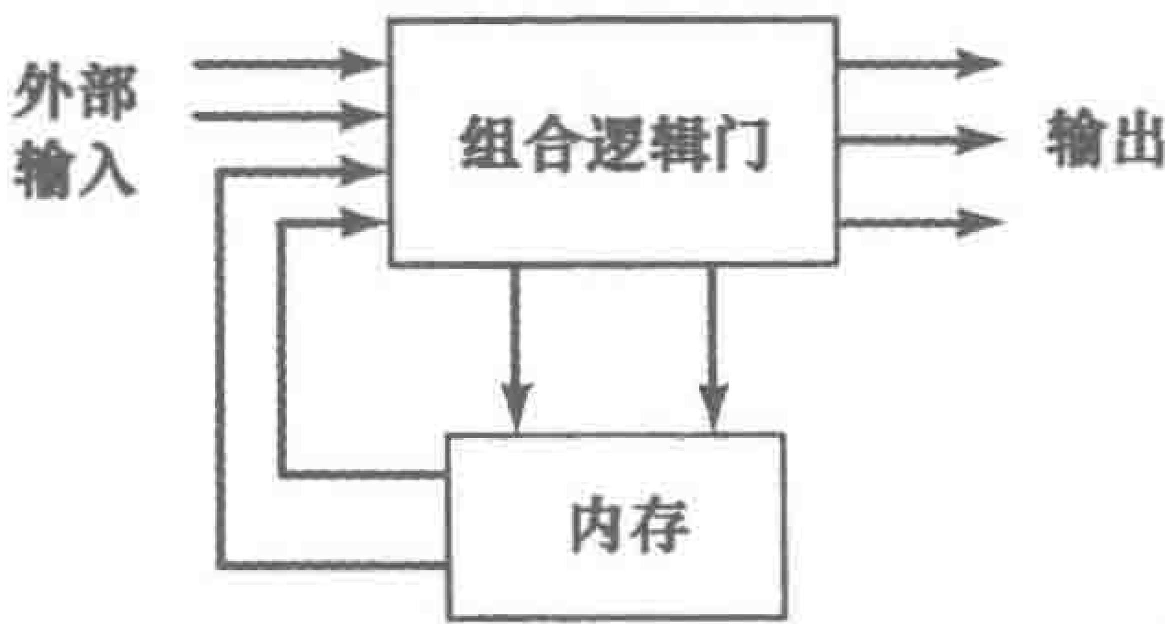


图 5-19 时序逻辑系统

图 5-19 给出了一个时序逻辑系统的基本形式。系统的组合逻辑部分首先从外部输入和存储器的输出接受逻辑信号，然后组合系统对这些输入信号进行运算产生输出信号。输出信号是外部输入和存储器存储的信息的函数。

5.4.1 触发器

126

触发器是一种基本的存储元件，由逻辑门组成，是时序逻辑器件。触发器有很多种，图 5-20a展示的是其中的一种，SR（set-reset）触发器由 NOR 门组成。如果初始状态时输出全部为 0，$S=0$，$R=0$，那么当将 *S* 由 0 置位为 1 时，NOR 门 2 的输出就是 0。这使得 NOR 门 1 的输入变为 0，也使得其输出变为 1。该反馈作为输入使得 NOR 门 2 的输入都为 1，没有进一步的变化。

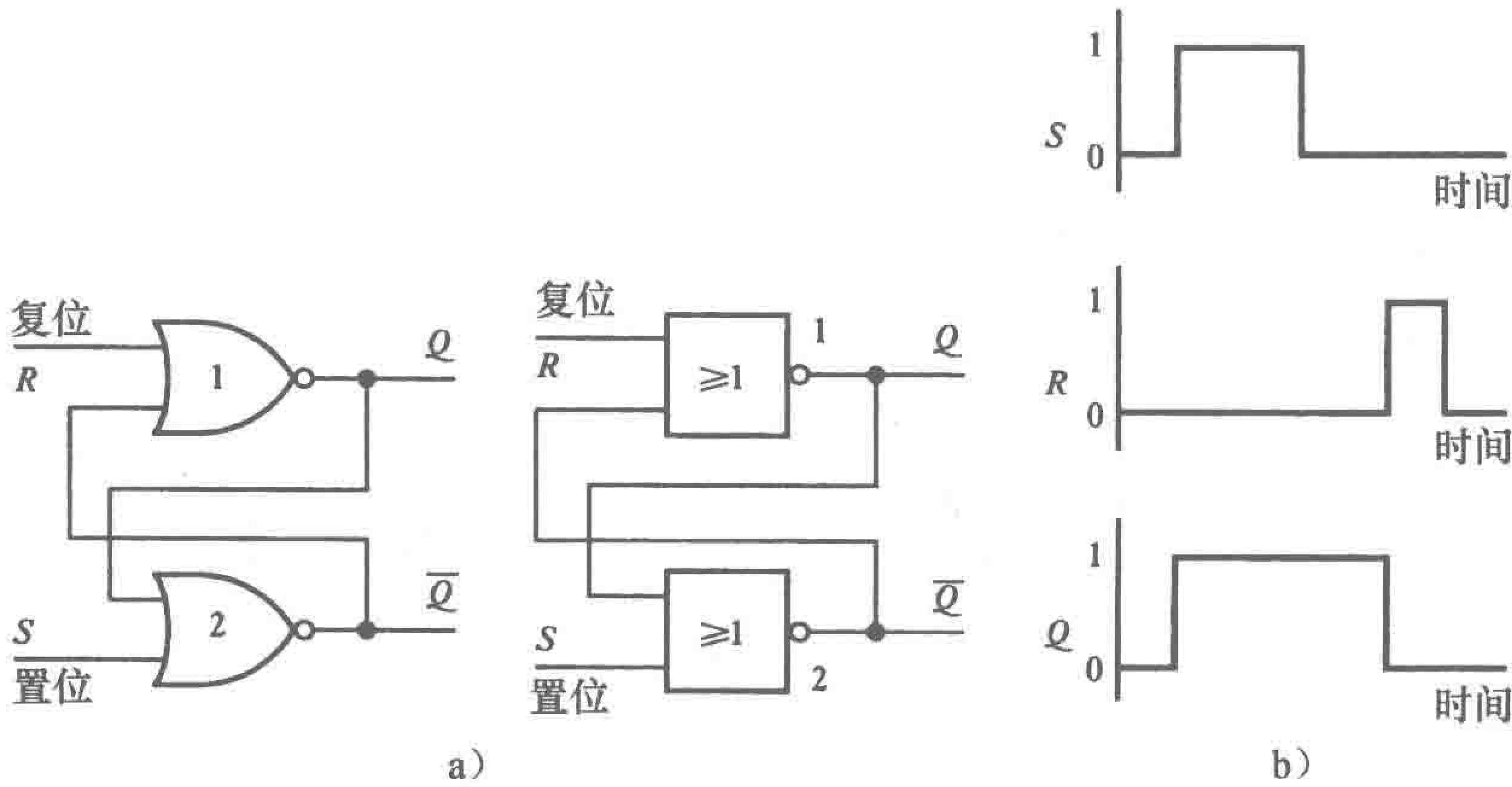
如果 *S* 从 1 变为 0，那么 NOR 门 1 的输出仍然保持 1，NOR 门 2 的输出保持 0。当输入 *S* 从 1 变为 0 时，输出没有任何改变。只有当 *R* 有了变化时，该状态才会有变化。它记忆下了其被设定的状态。图 5-20b 展示的是用方波脉冲表示 *S* 输入信号的时序图。

如果将 *R* 从 0 变为 1，而此时 *S* 是 0，那么 NOR 门 1 的输出变为 0，因此 NOR 门 2 的输出变为 1。复位触发器。那么 *R* 再变为 0，将对输出没有任何影响。

因此，当 *S* 设置为 1、*R* 设置为 0 时，如果之前输出 *Q* 是 0 就变为 1，如果之前是 1 就保持为 1。这就是置位状态，即使 *S* 变为 0 也不会改变输出。当 *S* 是 0 而 *R* 为 1 时，如果之前输出 *Q* 是 1 就复位为 0，如果之前是 0 就保持为 0。这是复位状态。输出 *Q* 在某一时刻的变化依赖于 *S*

与 R 的输入和上一个输出值。下面的状态表说明了这样的变化：

S	R	$Q_t \to Q_{t+1}$	$\overline{Q}_t \to \overline{Q}_{t+1}$
0	0	0→0	1→1
0	0	1→1	0→0
0	1	0→0	1→1
0	1	1→0	0→1
1	0	0→1	1→0
1	0	1→1	0→0
1	1	不允许	
1	1	不允许	



127

图 5-20　SR 触发器

需要注意的是，如果 S 和 R 同时为 1，则可能会造成状态的不稳定，所以这个输入条件禁止出现。图 5-21 展示的是 SR 触发器的简化方块符号。

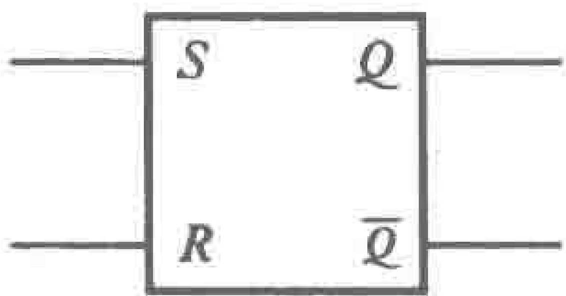


图 5-21　SR 触发器

在简单的报警系统中，当光束中断时，报警器便发出警报，即使光束不再中断，警报声仍然继续。图 5-22 是一种可能的实现系统。一个光电晶体管用做传感器并且如图 5-22 所示连接到电路中，当光照射它时会提供大约 0V 输入电压到 S，而当停止照射时就提供 5V 输入电压到 S 上。当光束中断时，S 变为 1，触发器的输出变为 1 且发出警报。输出将仍为 1，即使当 S 变为 0。只有当复位开关马上打开并且提供 5V 电压输入到 R 时，报警器才会停止。

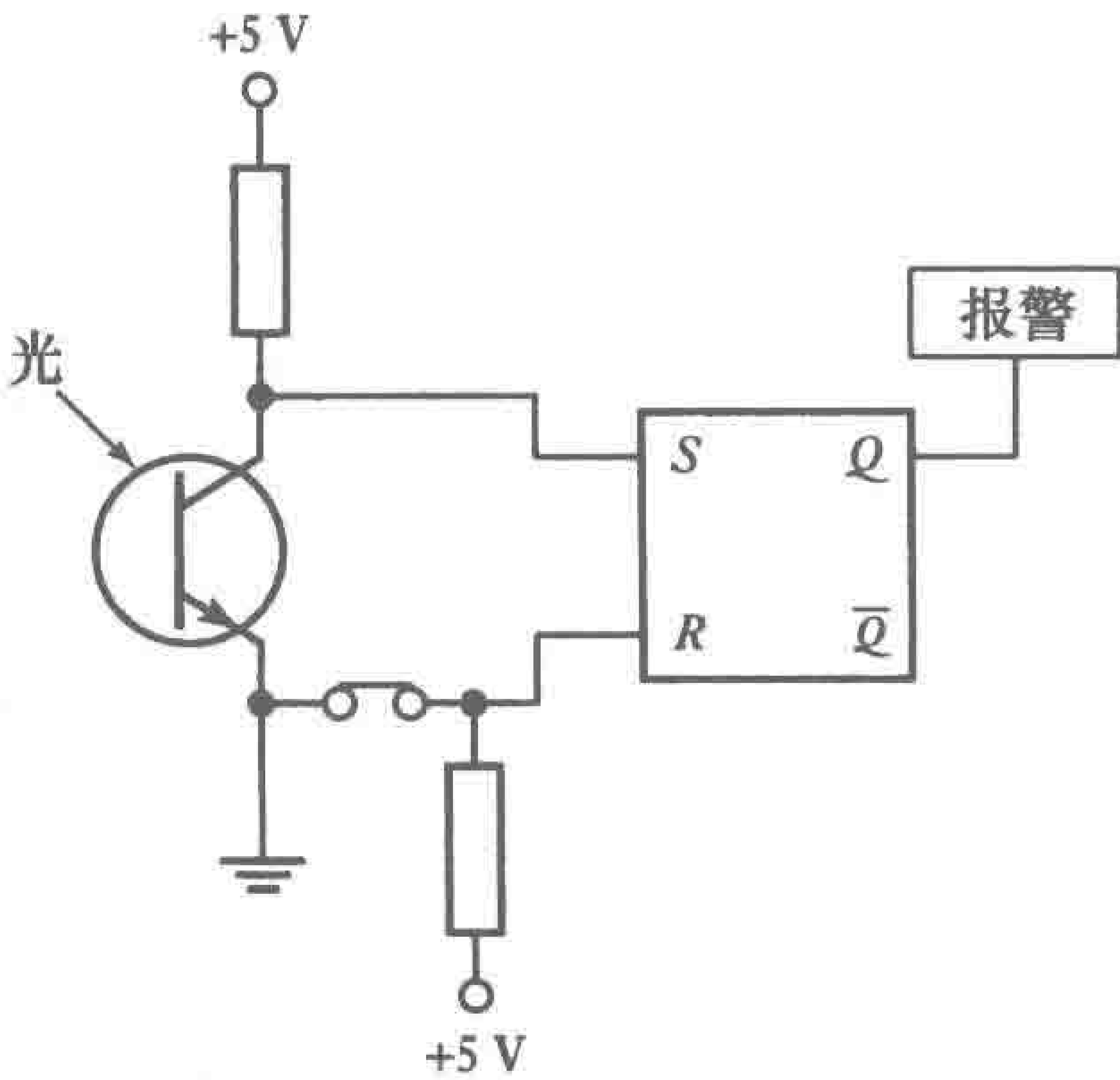


图 5-22　报警电路

5.4.2　同步系统

有时需要在特定的时间内进行置位和复位操作。在一个无时钟或**异步系统**中，逻辑门的输出在任何时刻完

全是由一个或多个输入的改变来决定的。在一个有时钟的系统或**同步系统**中，任何输出能够改变的时刻由时钟信号来决定。当一个矩形脉冲序列的时钟信号应用于整个系统时，输出是同步的。图5-23a展示的是**门控SR触发器**的原理。置位信号和时钟信号通过AND门把输入信号引到触发器的*S*端。只有在置位信号和时钟信号均为1时，它才能来到触发器的*S*端。同样，复位信号和时钟信号通过其他的AND门把输入信号引到*R*端。因此，置位和复位由时钟决定。图5-23b是其时序图。

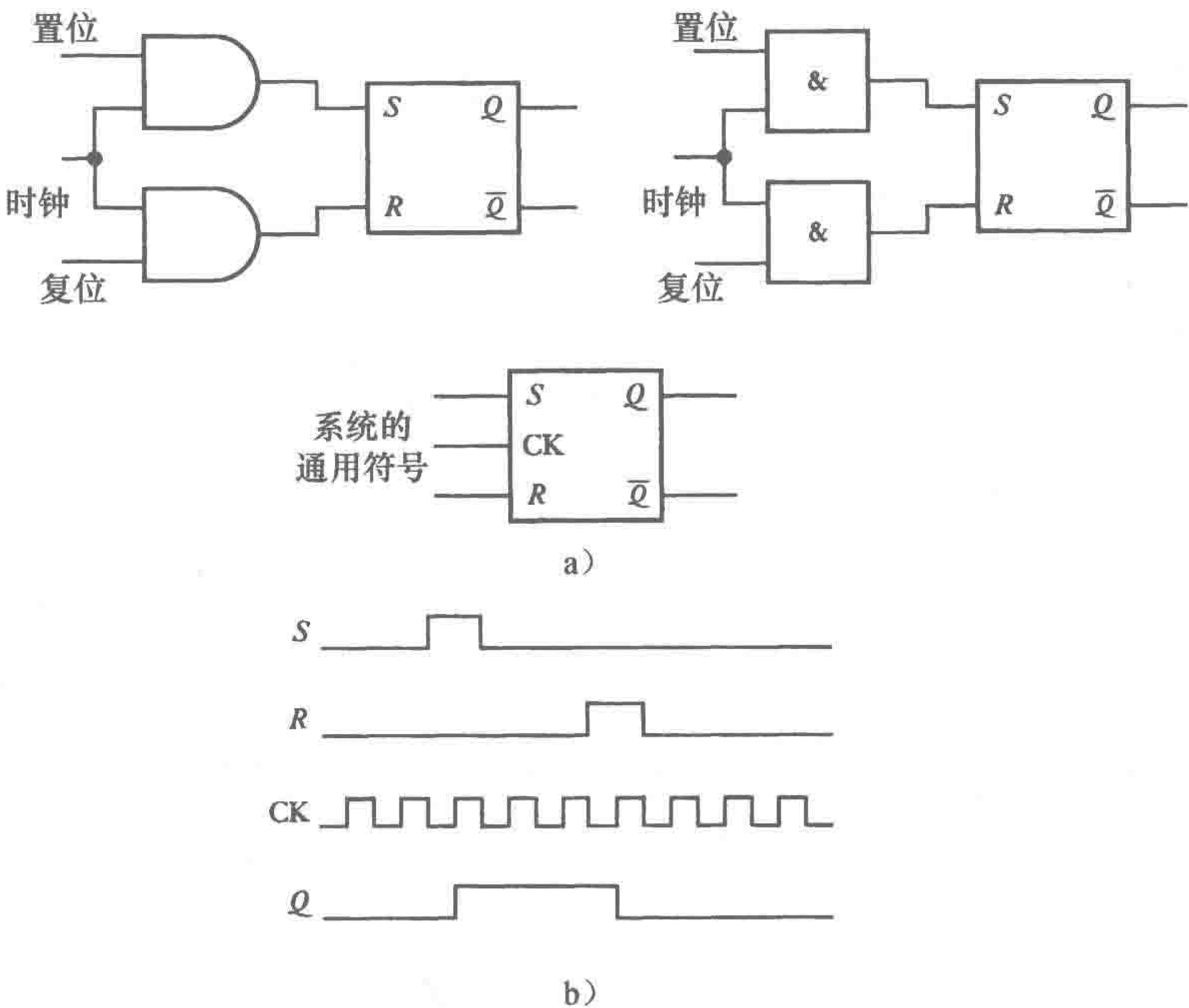


图5-23 门控SR触发器

5.4.3 JK触发器

在许多应用中，SR触发器在$S=1$且$R=1$时状态是不明确的，此时我们引入**JK触发器**（见图5-24）。这已成为使用非常广泛的触发器。

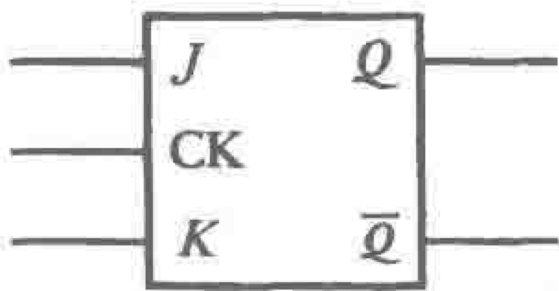


图5-24 JK触发器

下面是JK触发器的真值表。值得注意的是，不同于SR触发器真值表的地方仅仅在于两个输入均为1的情况：

J	K	$Q_t \to Q_{t+1}$	$\overline{Q}_t \to \overline{Q}_{t+1}$
0	0	0→0	1→1
0	0	1→1	0→0
0	1	0→0	1→1
0	1	1→0	0→1
1	0	0→1	1→0
1	0	1→1	0→0
1	1	0→1	1→0
1	1	1→0	0→1

作为这样一个触发器使用的一个说明，考虑当输入 A 变高电平，然后一段时间后 B 变高电平时输出才为高电平的要求。一个AND门可用于确定是否两个输入都是高电平，但无论哪个输入先变高电平，它的输出都变高电平。然而，如果输入 A 和 B 用于一个 JK 触发器，则为了使输出变高电平 A 必须先变高电平，然后 B 再变高电平。

128
~
129

5.4.4　D 触发器

数据或 **D 触发器**是有时钟的 SR 触发器或 JK 触发器，其中 D 端直接连接到 S 或 J 端，并且通过 NOT 门连接到 R 或 K 端（见图 5-25a）。D 触发器的逻辑符号上，R 和 K 端结合标注为 D。这意味着在时钟脉冲为 1 时，输出等于输入（见图 5-25b）。D 触发器的用途是确保输出准确地只在某一特定时刻呈现 D 输入。图 5-25c 为 D 触发器的逻辑符号。

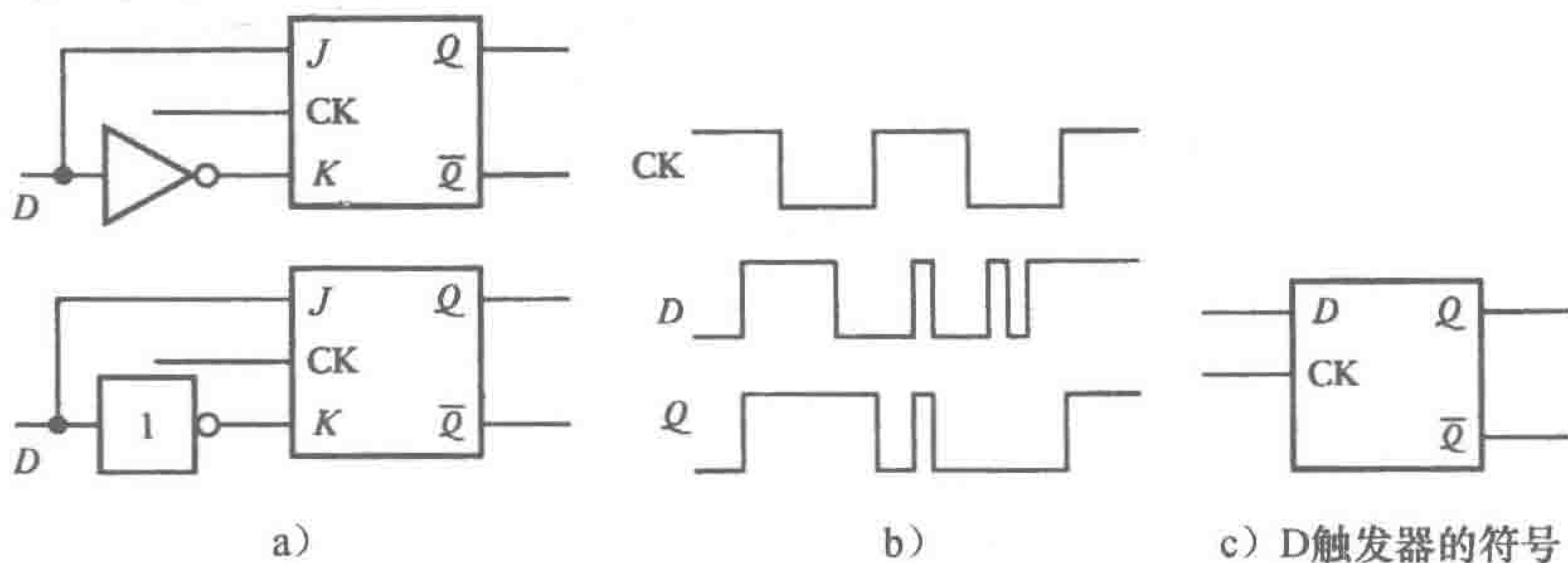


图 5-25　D 触发器

在上述的 D 触发器中，当时钟或者使能输入为高电平时，输出跟随输入 D 表示的数据。这时触发器称为"透明的"。如果使能端输入的状态由高电平转换成低电平，则输出Q将保留原数据电平。这时的数据转换状态称为**锁存**。D 触发器可用做集成电路。7475 就是一个例子，它包含 4 个 D 触发器。

7474 D 触发器和 7475 的主要区别在于前者是边沿触发器；一个封装里有两个这样的触发器。边沿触发 D 触发器中Q的变化仅取决于时钟脉冲的边沿，对于 7474 为上升沿，也就是由低电平转换为高电平的边沿（见图 5-26a）。边沿触发 D 触发器和图 5-25 中的 D 触发器的基本逻辑符号的主要区别在于靠近 CK 输入端处加了一个小三角形（见图 5-26b）。还有其他两个输入称为预置和清零。在清零时低电平将输出Q置 0，预置时低电平将输出Q置 1。

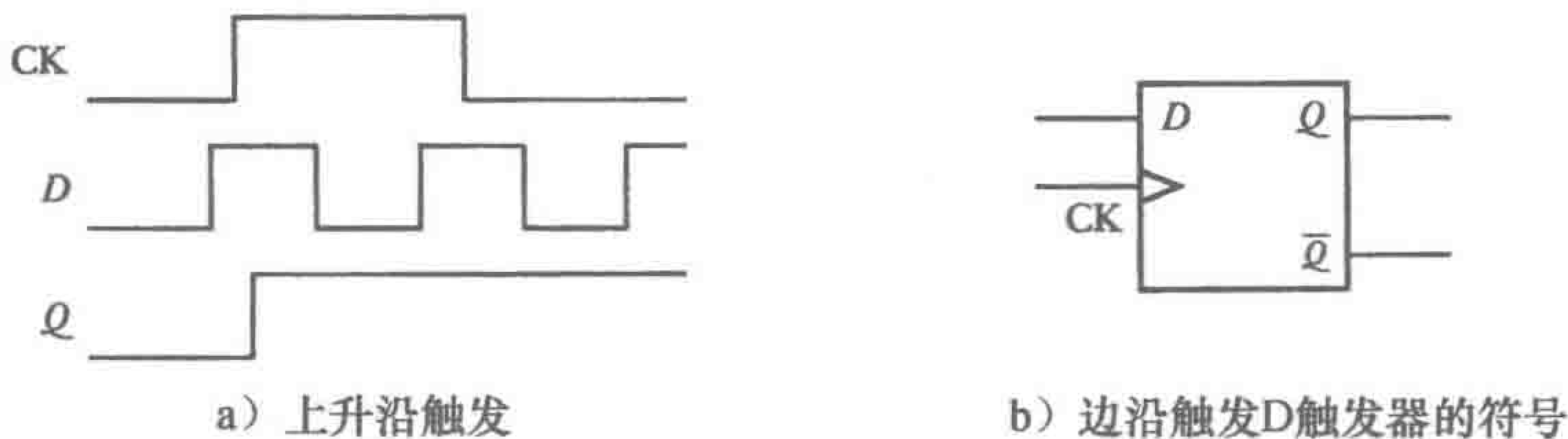


图 5-26　边沿触发 D 触发器

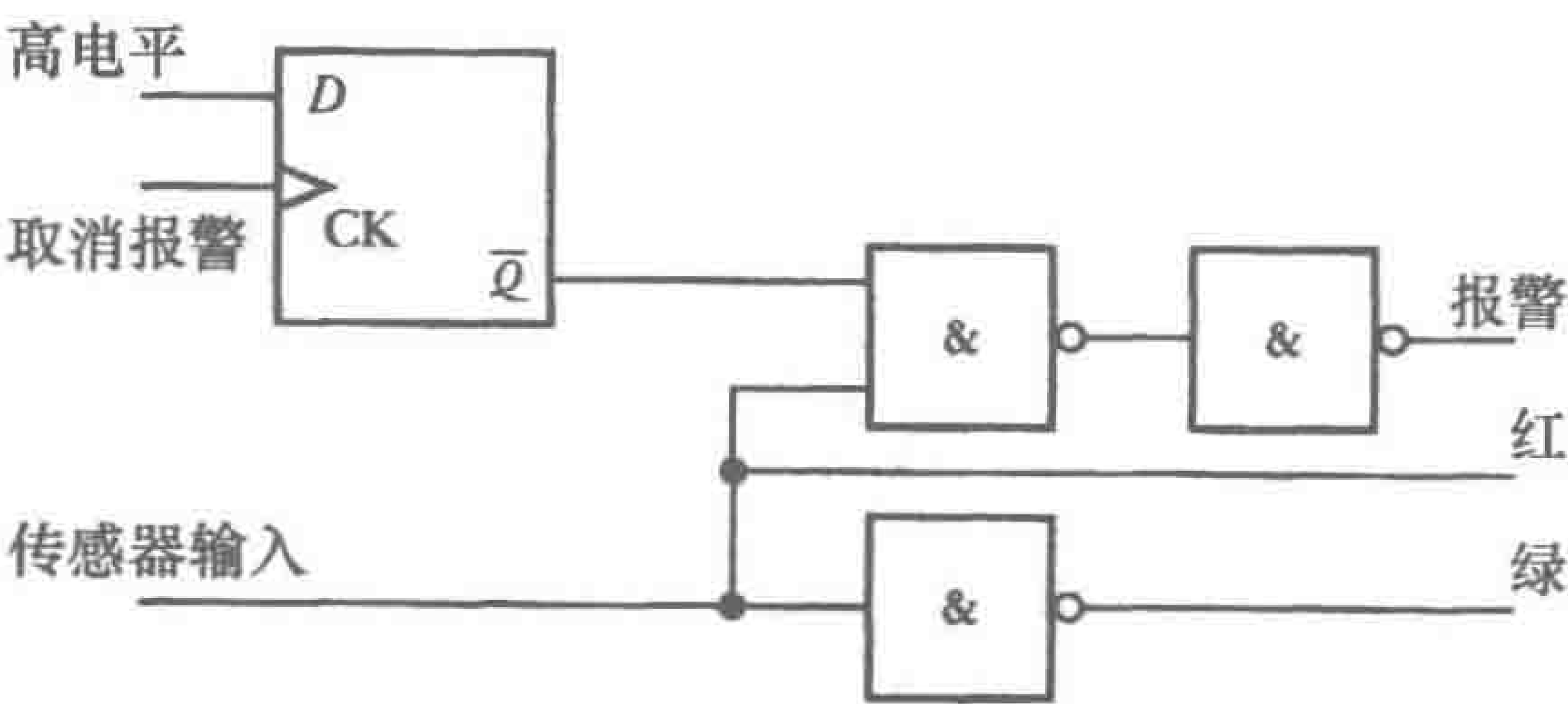


图 5-27　报警系统

图 5-27 显示系统传感器输入低电平时绿灯亮，其输入跳为高电平时红灯亮并发出警报。只要传感器输入为高电平，红灯就保持亮，但警报可以停止。这是一个用来监控温度的系统，当温度比安全限度低时，传感器和信号调理电路将给出低电平信号；当温度比安全限度高时，将给出高电平信号。当 CK 输入端为低电平且传感器为低电

平输入时，绿灯亮。当传感器输入转换成高电平时，绿灯熄灭，红灯亮并发出警报。给 CK 端输入高电平就可实现警报关闭，但是只要传感器输入为高电平，红灯就会一直亮。这样的系统由 7474 触发器和集成电路或 3 个 NAND 门的电路构成。

130

5.4.5 寄存器

寄存器是一组存储元件，可用来保存信息。它可由一组触发器来实现。一个触发器只能寄存 1 位二进制信号，即 0 或 1。图 5-28 展示的是由 D 触发器构成的 4 位寄存器。

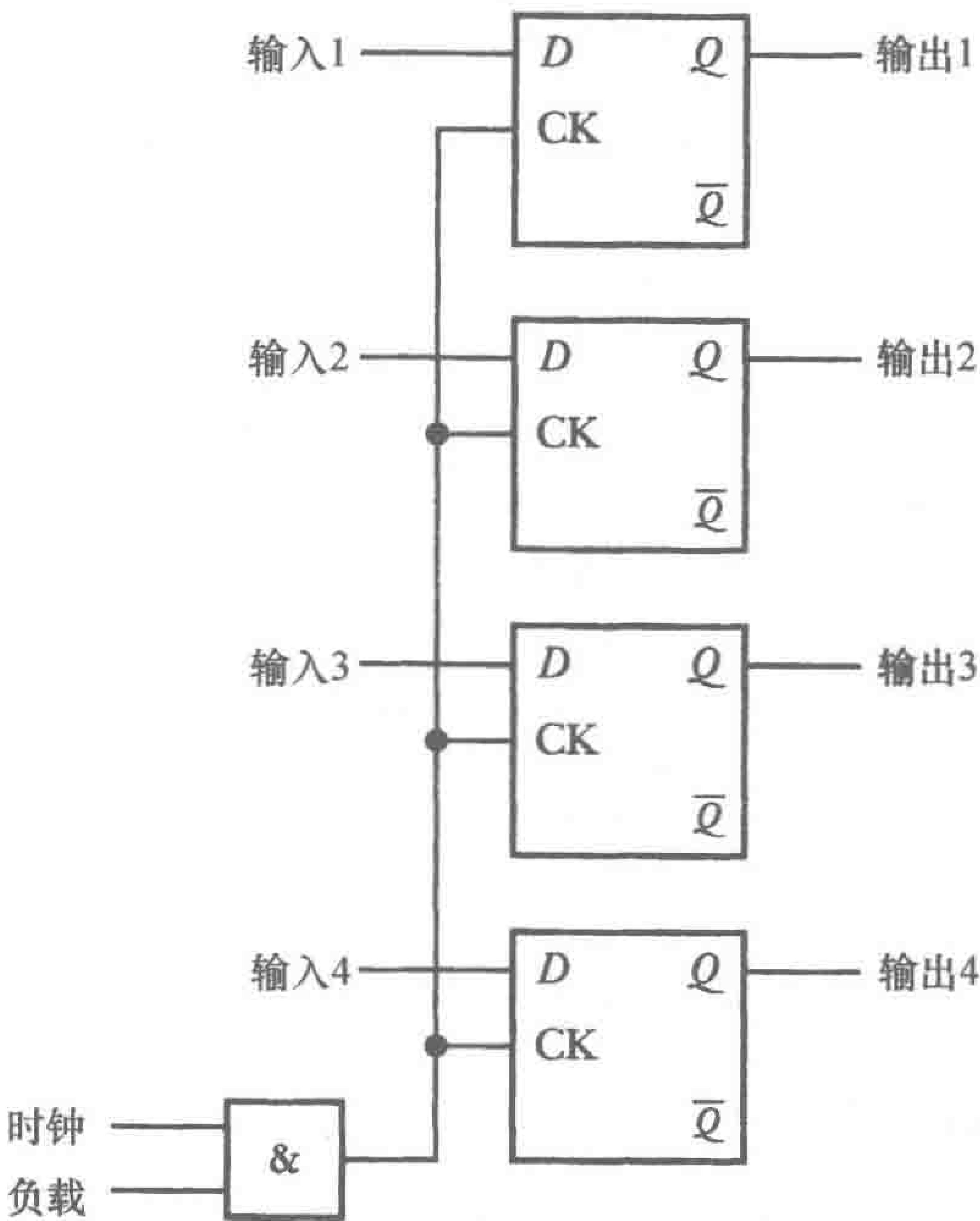


图 5-28 寄存器

当负载（load）信号为 0 时，没有时钟信号作用在 D 触发器上，所以触发器保持原态。当负载信号为 1 时，输入信号将改变触发器状态。仅当负载信号为 0 时，触发器才能保持原有状态。

5.4.6 555 定时器

555 定时器芯片广泛应用于数字电路，提供各种各样的定时任务。它由 SR 触发器和两个比较器提供的输入组成（见图 5-29）。每个比较器都有输入电压，输入电压由相同阻值的电位器串联提供。比较器 A 有一个相相输入电压 $V_{CC}/3$，B 的反相输入为 $2V_{CC}/3$。

555 定时器的一种应用是**单稳态多谐振荡器**（monostable multivibrator），这种电路可以在收到触发信号之后产生单脉冲，并且可以得到所需要的脉宽。图 5-30a 展示了这种应用的连接方法。初始状态时，输出为低电平，晶体管短路电容，两个比较器的输出均为低电平（见图 5-30b）。

131
~
132

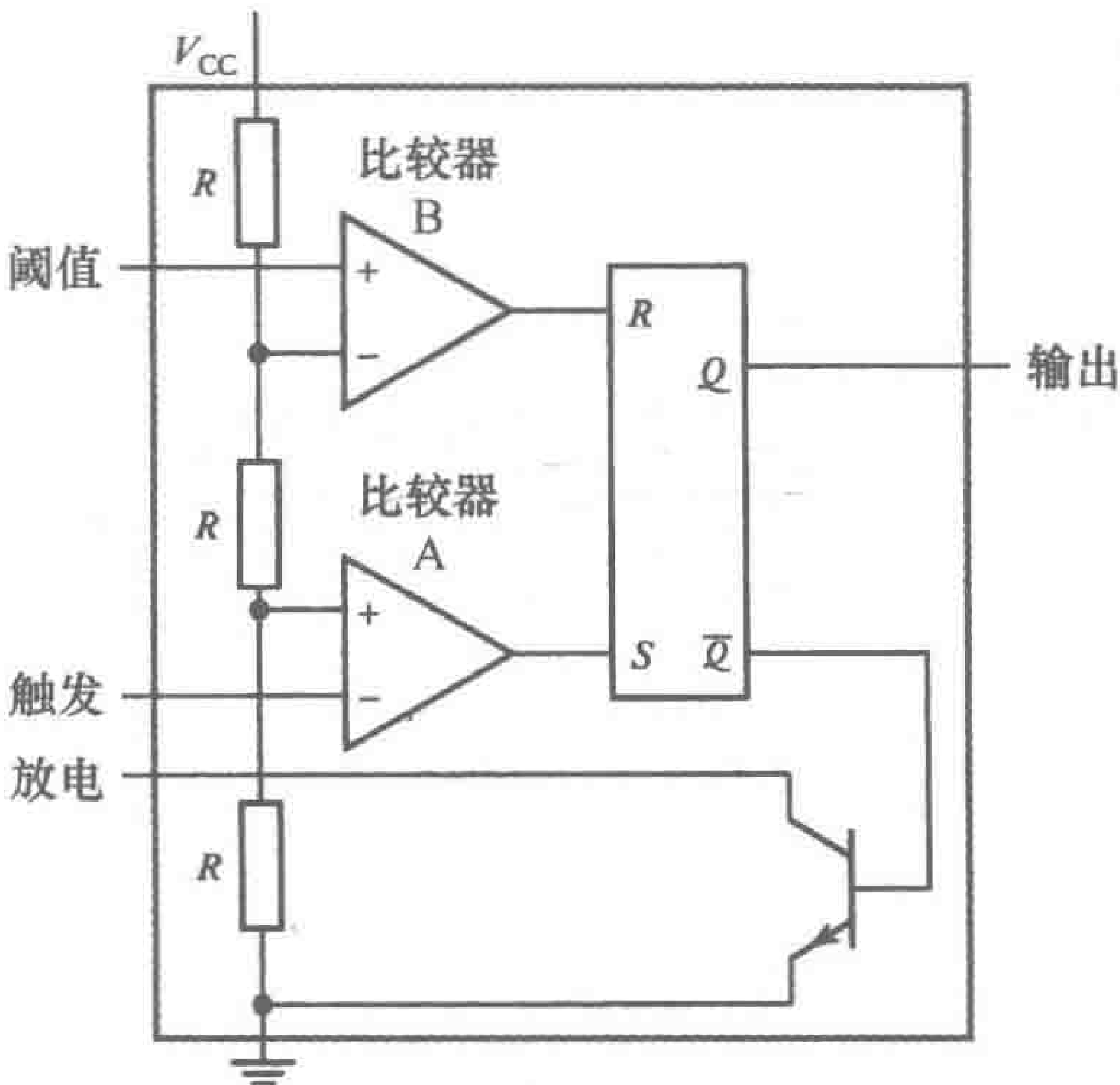


图 5-29 555 定时器

当触发脉冲低于 $V_{CC}/3$ 时，触发比较器为高电平并置位触发器。然后输出为高电平，晶体管截止，电容开始充电。当电容达到 $2V_{CC}/3$ 时，阈值比较器复位触发器，从而复位输出为低电平，使电容放电。如果输出为高电平，触发脉冲则无任何反应。脉冲的宽度即为电容充电至 $2V_{CC}/3$ 的时间，时间常量为 R_tC，由电阻对电容充电的一般关系式 $1.1R_tC$ 给出。在防盗系统中，当门打开时，如果户主在 30 秒内不输入必要的数字，防盗器就会响。如果图 5-30 所示电路的电容为 1μF，那么 R_t 的值为 $30/(1.1\times1\times10^{-6})=27.3\text{M}\Omega$。

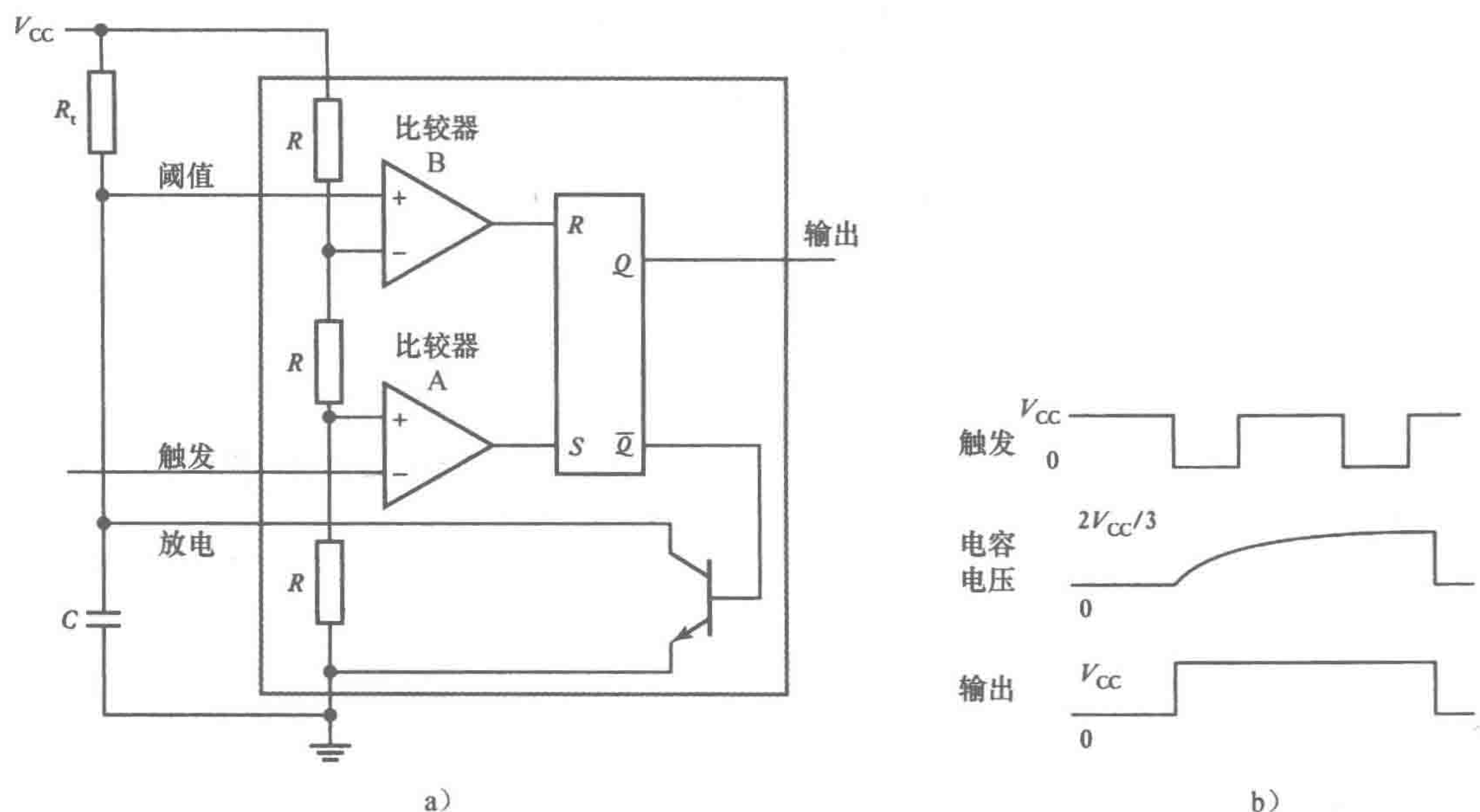


图 5-30 单稳态多谐振荡器

本章小结

在**组合逻辑系统**中，输出是由某一时刻输入的组合决定的，而不是由之前的输入决定的。当系统的输出需要由之前的输入决定时，就要引入**时序逻辑系统**。组合逻辑系统和时序逻辑系统的主要区别在于后者必须有某种形式的存储器。

常见的逻辑系列有：**晶体管晶体管逻辑**（Transistor-Transistor-Logic，TTL）、**互补金属氧化物半导体**（Complementary Metal-Oxide Semiconductor，CMOS）和**射极耦合逻辑**（Emitter-Coupled Logic，ECL），由逻辑电平、噪声容限、扇出系数、拉电流和灌电流、传播延迟时间和功率损耗来区分。

译码器是一种逻辑电路，它接受输入信号，判断其对应哪个数，并激活对应那个数的输出端。

触发器是一种基本存储元件，由逻辑门电路组成，是时序逻辑器件。

寄存器是由一系列的存储元件组成的，可用来保存信息。

555 定时器是由 SR 触发器和两个比较器提供的输入组成的。

习题

5.1 解释使用哪种逻辑门能控制下列情况：

(a) 地铁站里自动售票机的自动售票。

(b) 机床的安全锁系统。

(c) 热水器在达到某一温度（如 60℃）时关闭开关，循环泵自动关闭。

133 (d) 当升降机的门关闭并且按下所要去的楼层按钮时，发出信号使升降机开始运动。

5.2 对于图 5-31 所示的 A 和 B 的时序信号，当 A 和 B 输入到 AND 门和 OR 门时，P 和Q哪一个是对应的输出信号？

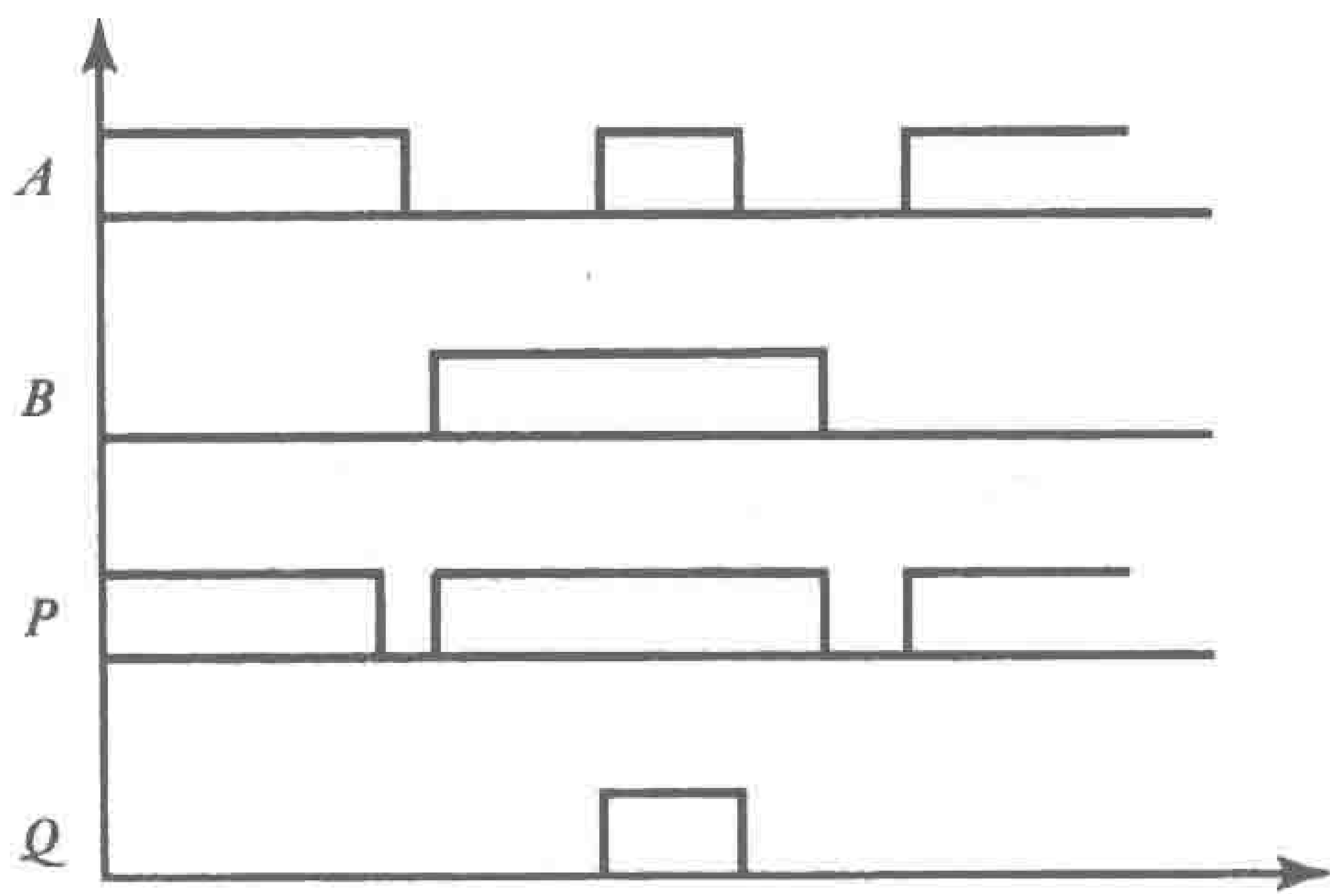


图 5-31 习题 5.2

5.3 将连续时序脉冲时钟信号连接到逻辑门上，当且仅当使能信号也输入给该门时才有输出，请问应该选择哪种逻辑门？

5.4 输入 A 直接连在双输入 AND 门上，输入 B 先接 NOT 门然后接在 AND 门上，在输入 A 和 B 满足什么条件时，AND 门会输出 1？

5.5 将图 5-32a 中的输入信号 A 和 B 应用于图 5-32b 的门系统，画出 P 和 Q 的输出结果的时序图。

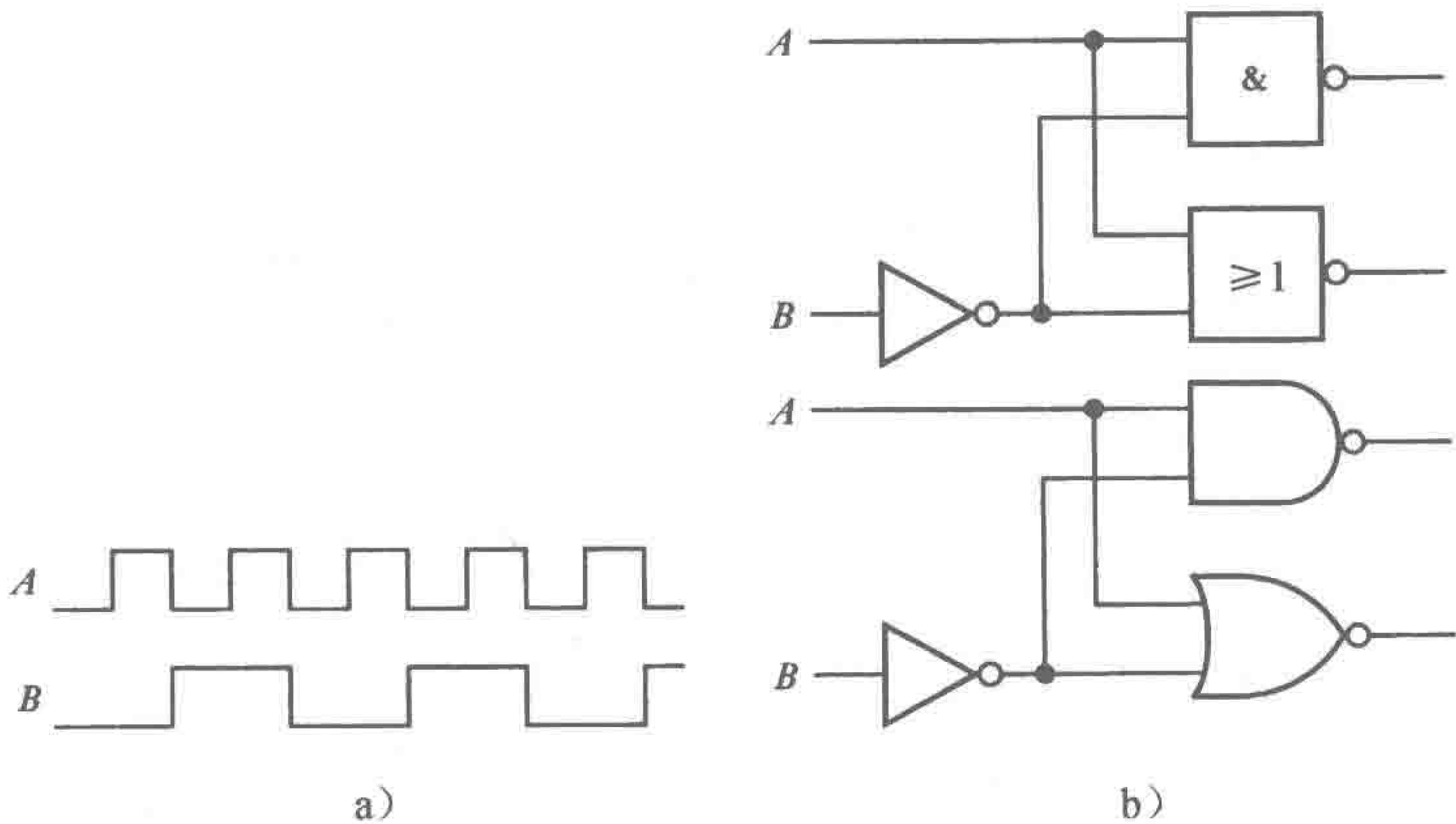


图 5-32 习题 5.5

5.6 图 5-33 是 SR 触发器的 S 和 R 输入端的时序图。试画出输出端 Q 的图形。

5.7 分析图 5-34 给出的 SR 触发器的逻辑功能。

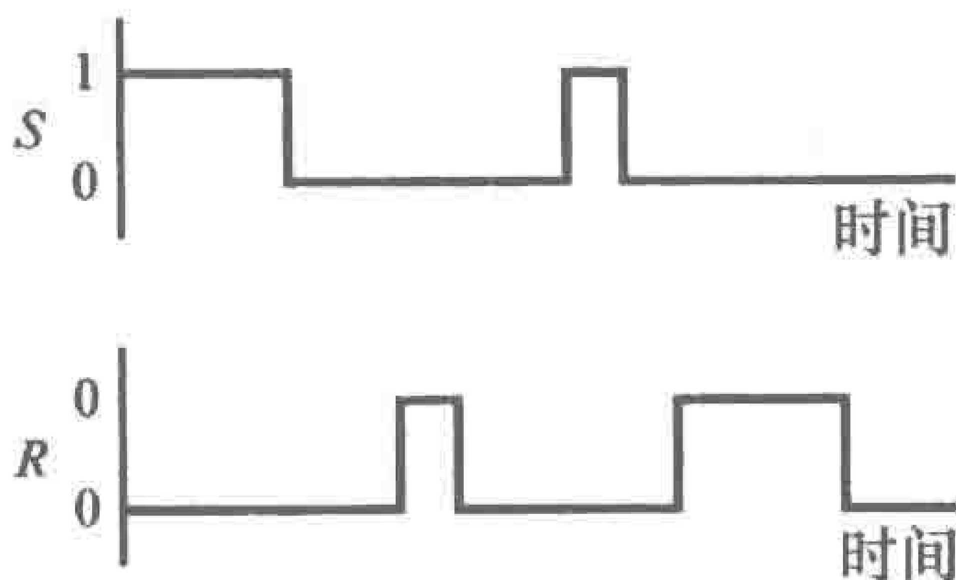


图 5-33 习题 5.6

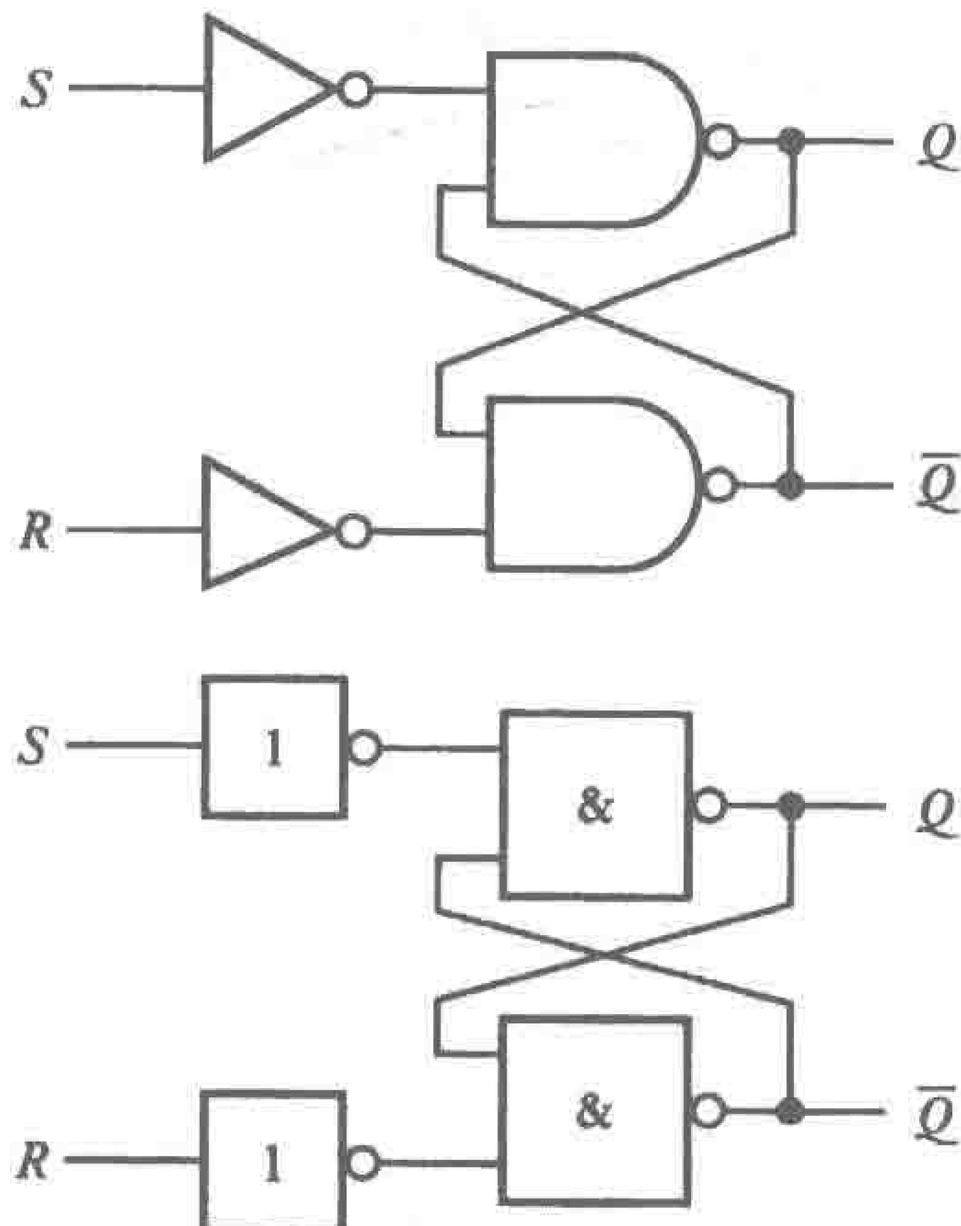


图 5-34 习题 5.7

第6章 数据显示系统

本章目的

通过本章的学习，读者将能够：

- 解释测量系统中的负载效应问题。
- 认识常用的数据表达要素，并能够描述各要素的特点。
- 解释磁和光的记录原理。
- 解释显示原理，特别是 LED 七段和点阵的显示原理。
- 描述数据采集系统和虚拟仪器的基本要素。

6.1 显示方式

本章将介绍数据在发光二极管（LED）或者在计算机显示器上是如何显示的，以及在计算机硬盘或 CD 中是如何存储的。

测量系统包括三个要素单元：传感器单元、信号调理单元和数据显示单元（参见 1.4 节）。可用于数据显示的元件包括很多种类。通常将这些元件分为两大类：指示器和记录器。**指示器**显示的是传感变量的瞬时值，而**记录器**则记录输出信号在某一段时间内的信息，并将这些信号自动永久保存。

本章可以是前几章中与测量系统相关的传感器、信号调理和显示的实现。因此，本章给出了一个完整的测量系统的例子。

6.1.1 负载效应

当组装任何一套测量系统时，负载都是必须重视的一方面，也就是要考虑把新的负载连接在系统任何一个元件的输出终端上时带来的影响。

测量回路电流时，在回路中串联电流表会改变回路中的电阻，于是电流也就相应地改变了，实际上这时候测量的是改变后的电流。如果把电压表跨接在电阻两端，就相当于电压表和电阻并联，如果电压表的内阻不是理想地远大于电阻的阻值，那么电阻两端的电压就会产生明显的变化，于是这时候测量的电压值也就不是实际的电压。测量电压时实际上已经改变了电阻两端的电压，这就叫作负载效应。

在测量过程中，当把一个元件和另一个元件连接起来的时候，产生的负载效应也会影响元件的特性。例如，一个测量系统包括一个传感器、一个放大器和一个显示元件（见图 6-1）。该传感器开路输出电压是 V_s，电阻是 R_s。放大器输入电阻是 R_{in}，这是跨接在传感器两端的负载。因此放大器得到的输入电压 V_{in} 是：

$$V_{in} = \frac{V_s R_{in}}{R_s + R_{in}}$$

如果放大器电压增益为 G，那么它的开路输出电压就是 GV_{in}。如果放大器输出电阻是 R_{out}，那么放大器输出电压就会被分压，电阻 R_d 的显示元件两端电势差 V_d 是

$$V_d = \frac{GV_{in}R_d}{R_{out} + R_d} = \frac{GV_s R_{in} R_d}{(R_{out} + R_d)(R_s + R_{in})} = \frac{GV_s}{\left(\frac{R_{out}}{R_d} + 1\right)\left(\frac{R_s}{R_{in}} + 1\right)}$$

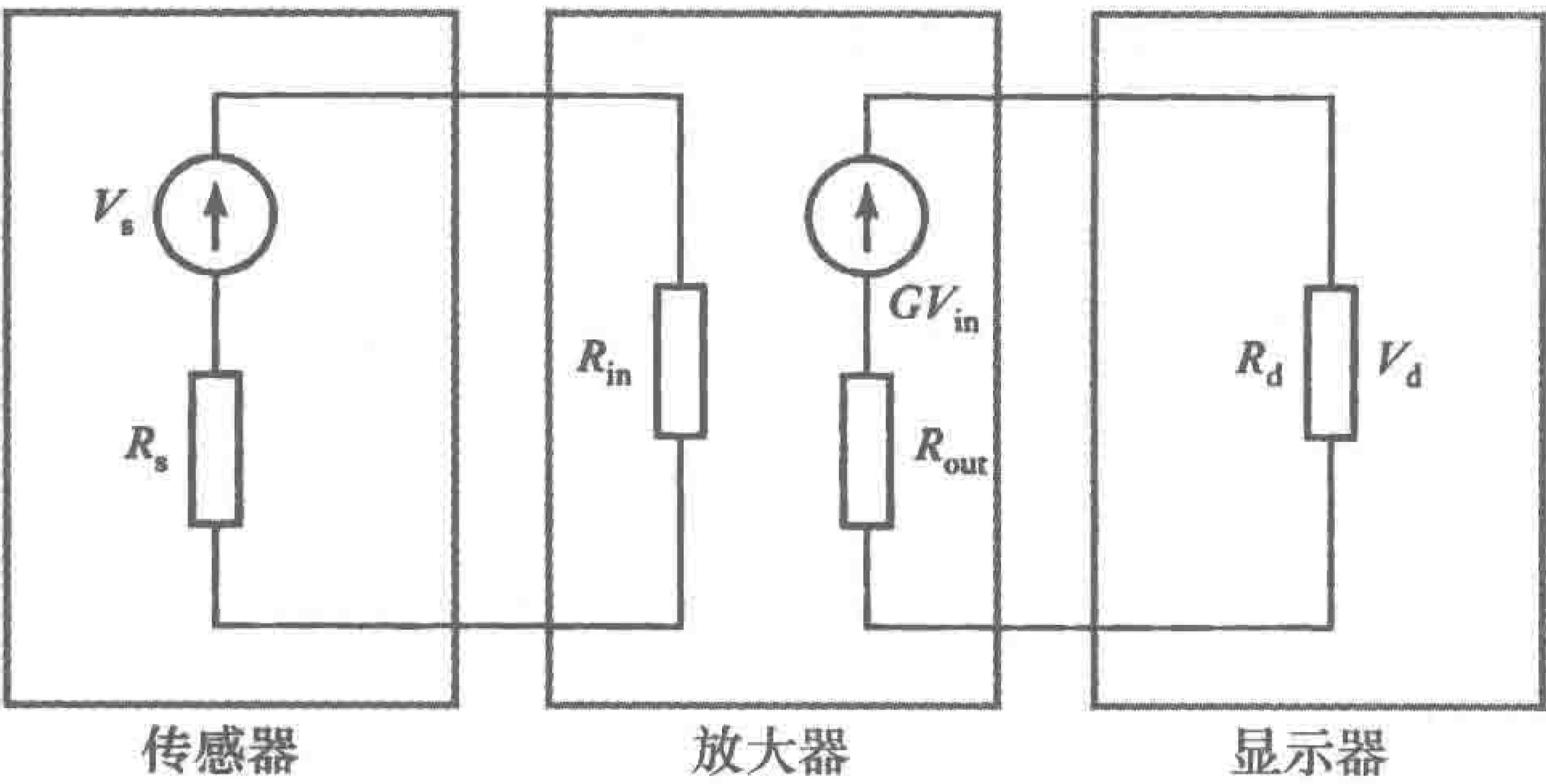


图 6-1 测量系统负载

因此，如果要忽略负载效应，就需要 $R_d \gg R_{out}$ 和 $R_{in} \gg R_s$。

6.2 数据显示元件

本节将简要举例概述常用的数据显示元件。

6.2.1 模拟仪表和数字仪表

动圈式仪表是一种指针移动式的模拟仪表。基本组成是一个直流微安表，其中带有分流器、乘法器和整流器，它们被用来转换模拟量到直流电流的其他范围，从而测量交流电流、直流电压和交流电压。对于交流电流和电压，该仪器的频率限制在 50Hz ~ 10kHz。这种仪表的精度取决于很多因素，比如温度、周围磁场、有色金属材料、安装方式和轴承摩擦，以及在制造过程中刻度标记的不准确度等。此外还有抄表时产生的错误，例如视差（当我们选取的视角变化时，读数也会有所不同）。总体误差大约在 ±0.1% ~ ±0.5% 的数量级。对于一个动圈式仪表，通常需要几秒钟就可以达到一个稳定的偏转角度。如果仪表的输入电阻很小，就可能出现负载效应问题。

137

数字电压表通过显示一串数字来显示读数。这种形式的显示消除了视差和插值误差，可以提供高达 ±0.005% 的精度。数字电压表基本上可以看成是将一个采样和保持单元连接到模数转换器（ADC），其输出由计数器来计数（见图 6-2）。它的输入电阻高达 10MΩ 数量级，所以它所带来的负载效应远远没有动圈式仪表那么明显。因此，如果一个数字电压表规定采样频率约为 5，那么这意味着每 0.2s 要对输入电压采样。在这 0.2s 内，它对数据进行处理并提供读数。因此，如果输入电压在 0.2s 内产生明显变化，那么该电压表读数可以说是错误的。低成本的数字电压表通常采样频率为每秒 3 次，输入阻抗为 100MΩ。

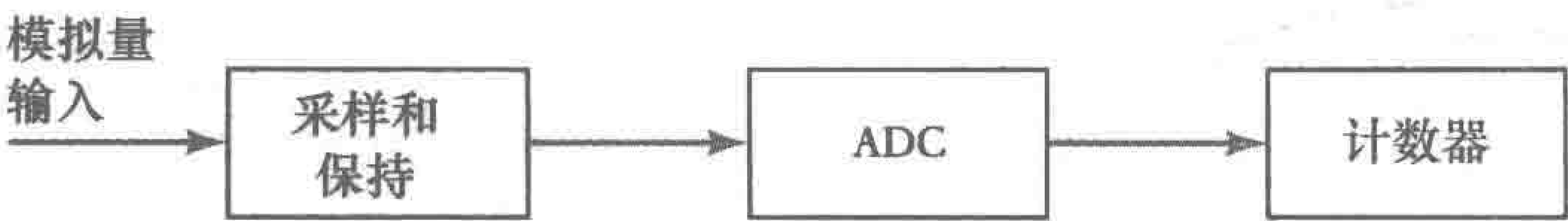


图 6-2 数字电压表原理

6.2.2 模拟图记录仪

模拟图记录仪采用极细纤维墨水笔将数据记录在纸上；使用压力指针在纸上加压碳带；通过使用热敏纸张，当加热指针移动时，热敏纸张会变色；通过紫外光束落在对紫外线敏感的纸上；由钨丝触针在特殊涂层的纸表面移动；一层薄铝覆盖在彩色染料上，当放电去除铝层后露出染料。但在许多应用中，它们已经被虚拟仪器取代了（见 6.6.1 节）。

6.2.3 阴极射线示波器

阴极射线示波器是一种电压测量仪，能显示极高频信号。一般的仪器可以对 10MHz 的信

138 号做出响应，而更专业的仪器可以响应高达1GHz的信号。双踪示波器可以分开同时观察两路信号，而存储示波器即使在信号停止时，轨迹仍然停留在屏幕上，除非使用擦除功能将其清除。数字存储示波器可以将输入信号数字化，并存储在内存中。示波器可以分析并重建信号，然后显示在屏幕上。如果要永久记录信号轨迹，则可把专用相机直接连接到示波器上。

通用的示波器可能有垂直偏转现象，即Y偏移，灵敏度在每刻度5mV~20V之间。为了从直流高电压中分辨出交流分量，可以把一个隔直电容器切换到输入线。当放大器工作在交流方式时，其带宽范围通常从2Hz~10MHz；在直流方式时，则是从直流到10MHz。该Y输入阻抗通常是1MΩ，并与20pF的电容并联。当一个外部电路连接到Y输入时，负载和干扰可能会扭曲输入信号。为了减少干扰可以使用同轴电缆，同轴电缆的电容和任何连接到电缆上的探针在低频情况下对示波器的输入阻抗会引入一个相对低的阻抗，因此会引入明显的负载效应。多个探针用于连接到输入电缆，目的是提高输入阻抗，避免这种负载效应问题。经常使用的无源电压探头是10:1的衰减器（见图6-3），在探针的顶部有一个9MΩ电阻和可变电容。然而，这不仅降低了电容性负载效应，也降低了电压的灵敏度，所以经常使用场效应晶体管（FET）的有源电压探针。

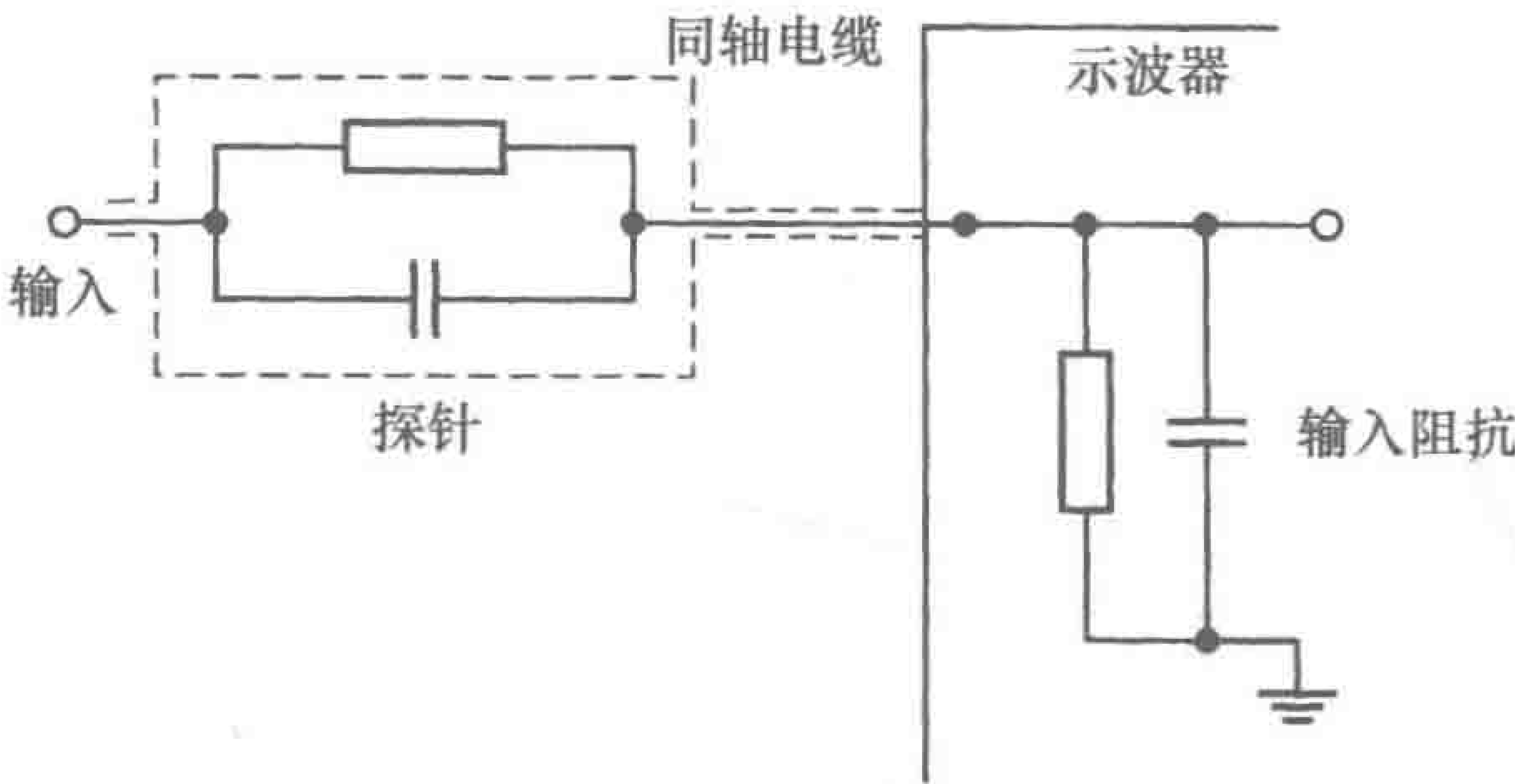


图6-3　无源电压探针

6.2.4　视觉显示单元

目前，越来越多地使用视觉显示单元（VDU）来显示输出数据。用一系列水平扫描线中电子束的移动点，可形成阴极射线管屏幕上的图像。每一帧图像由可变光强的电子束顺序地一行接着一行连续扫描而成，这种光栅显示方式称为**逐行扫描**（见图6-4a）。为了减少闪烁，把每一帧图像通过从上到下两次扫描来完成。第一次扫描所有的奇数线，第二次扫描所有的偶数线。这一技术称为隔行扫描（见图6-4b）。139

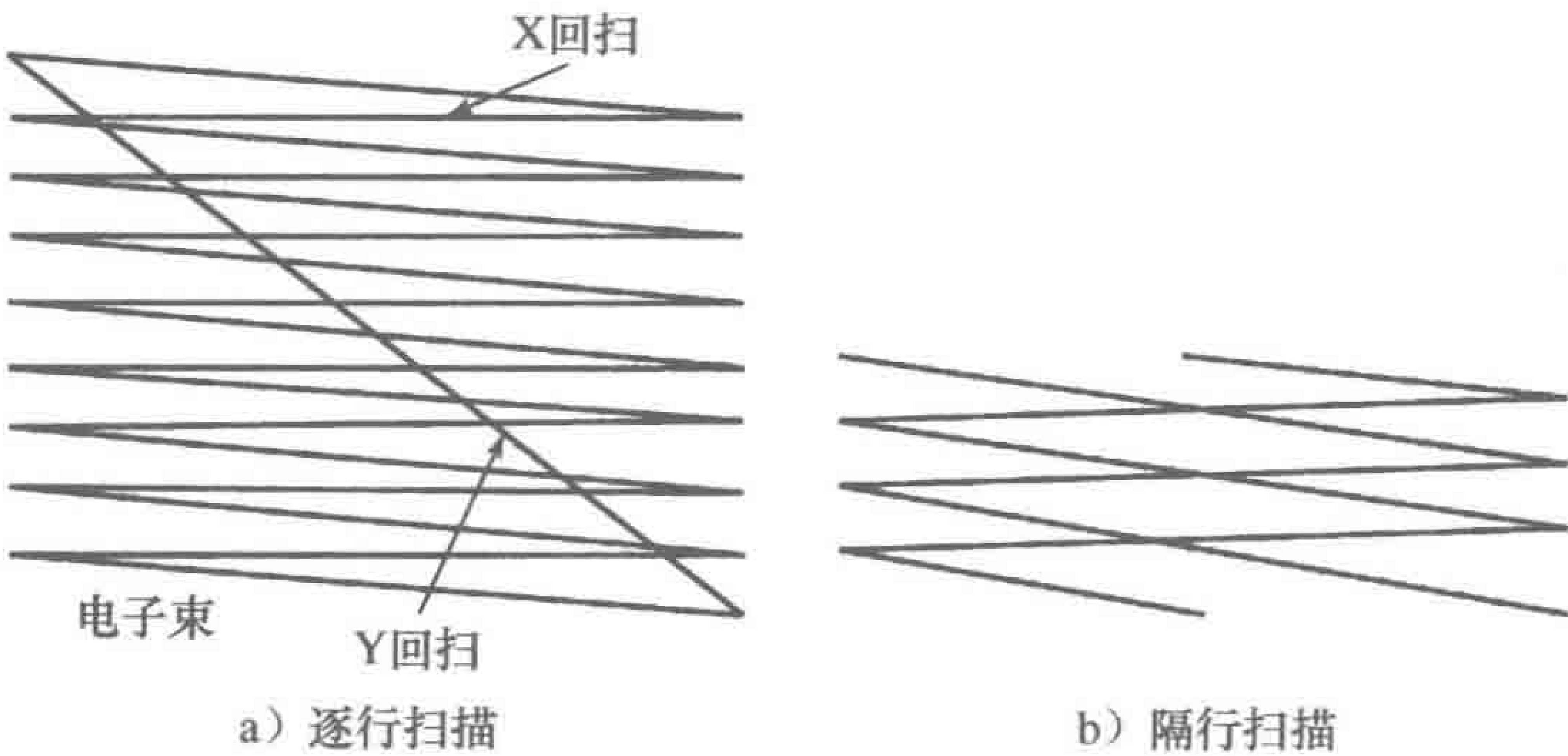


图6-4　逐行扫描和隔行扫描

VDU的屏幕涂有大量荧光点，这些点形成**像素**。像素是每个显示装置上的最小可寻址点。通过有选择地点亮这些点，可产生相应的文字或图案。图6-5说明了如何通过“之”字形电子束运动轨迹，在屏幕上显示出了7×5的矩阵。输入到显示单元的数据格式通常使用**ASCII码（美国信息交换标准码）**。这是一种7位代码，因此可以用来代表$2^7=128$个字符。它覆盖了所有的标准键盘的字符和一些控制功能，例如，RETURN用来表示从一行的后面返回到下一行的

开始。表 6-1 给出了一个简略的代码列表。

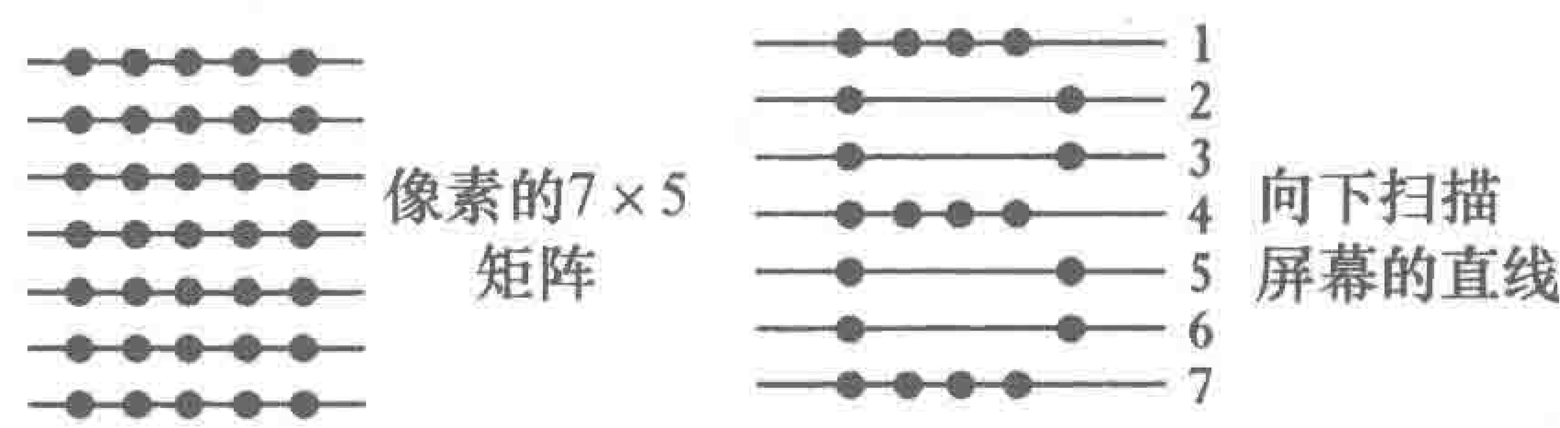


图 6-5　通过有选择地点亮产生字符

表 6-1　ASCII 码

字符	ASCII 码	字符	ASCII 码	字符	ASCII 码
A	100 0001	N	100 1110	0	011 0000
B	100 0010	O	100 1111	1	011 0001
C	100 0011	P	101 0000	2	011 0010
D	100 0100	Q	101 0001	3	011 0011
E	100 0101	R	101 0010	4	011 0100
F	100 0110	S	101 0011	5	011 0101
G	100 0111	T	101 0100	6	011 0110
H	100 1000	U	101 0101	7	011 0111
I	100 1001	V	101 0110	8	011 1000
J	100 1010	W	101 0111	9	011 1001
K	100 1011	X	101 1000		
L	100 1100	Y	101 1001		
M	100 1101	Z	101 1010		

140

6.2.5　打印机

打印机可以把数据记录在纸上。打印机分为点阵打印机、油墨/气泡喷墨打印机和激光打印机等类型。

点阵打印机的打印头（见图 6-6）由 9 或 24 根垂直排列的针组成。当控制每个针的电磁铁运动时，推动针撞击色带，进而在打印纸上留下一小点墨迹。字符是通过在纸上左右水平移动打印头，并控制相应的针撞击来打印的。

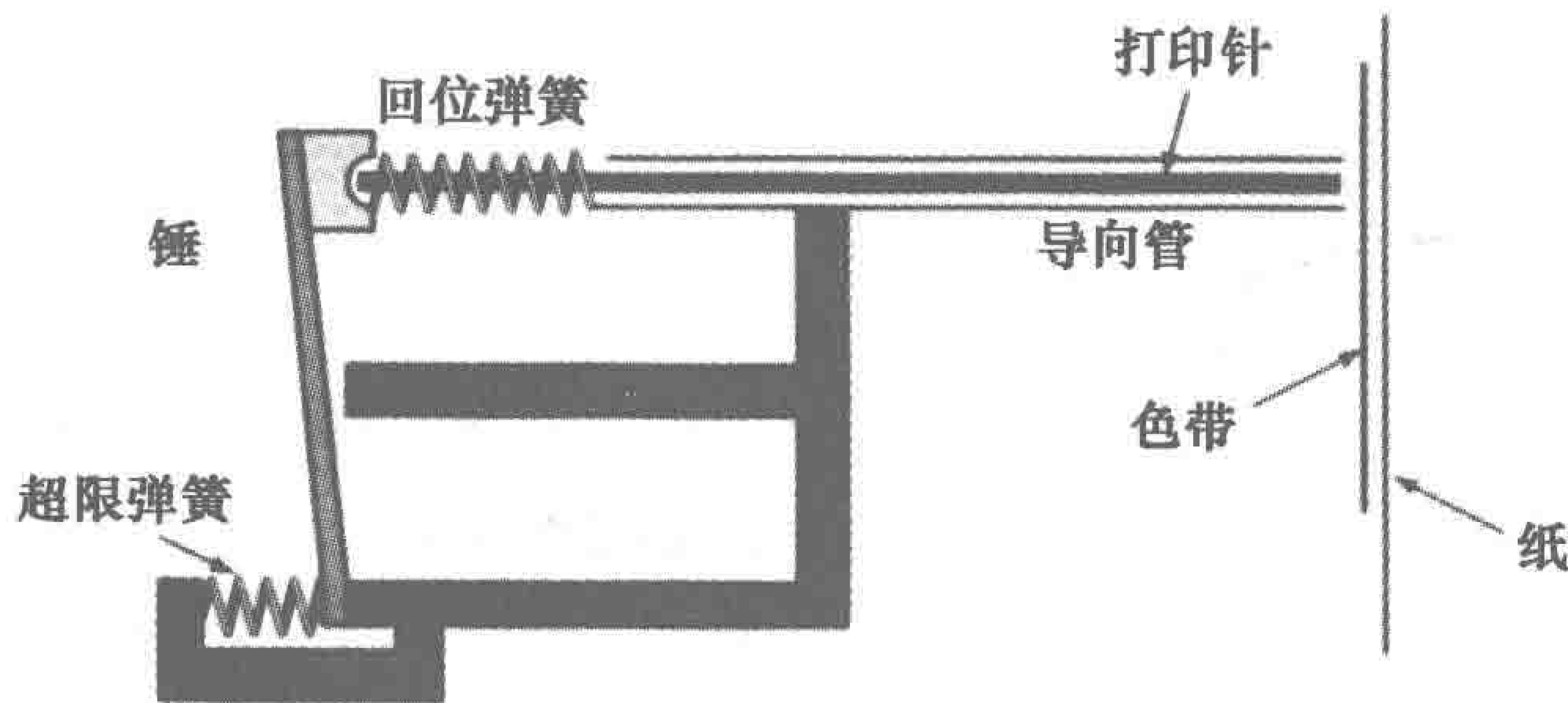


图 6-6　点阵打印头机构

喷墨打印机有多个固定直径的小喷嘴，当以固定频率喷射出微小导电墨滴时可实现打印。在 100kHZ 左右的压电晶体振动脉冲激励下，可使一系列的墨滴从喷嘴细管里喷出来（见图 6-7）。另一种方法是用一个小型加热器使打印头毛细管内的气泡膨胀，从而推出墨滴（见图 6-8）。还有一种喷墨打印机，当被充电的墨滴形成后，在电极间电场电压的作用下，墨滴偏

转而实现打印。而另一种喷墨打印机是通过控制喷嘴阵列中的每个喷嘴的开关来实现打印的。喷墨打印机可以通过三种不同颜色的墨水来实现彩色打印（也有三种以上颜色的情况）。通过在每英寸内产生超过600点的分辨率，可得到精细的打印效果。

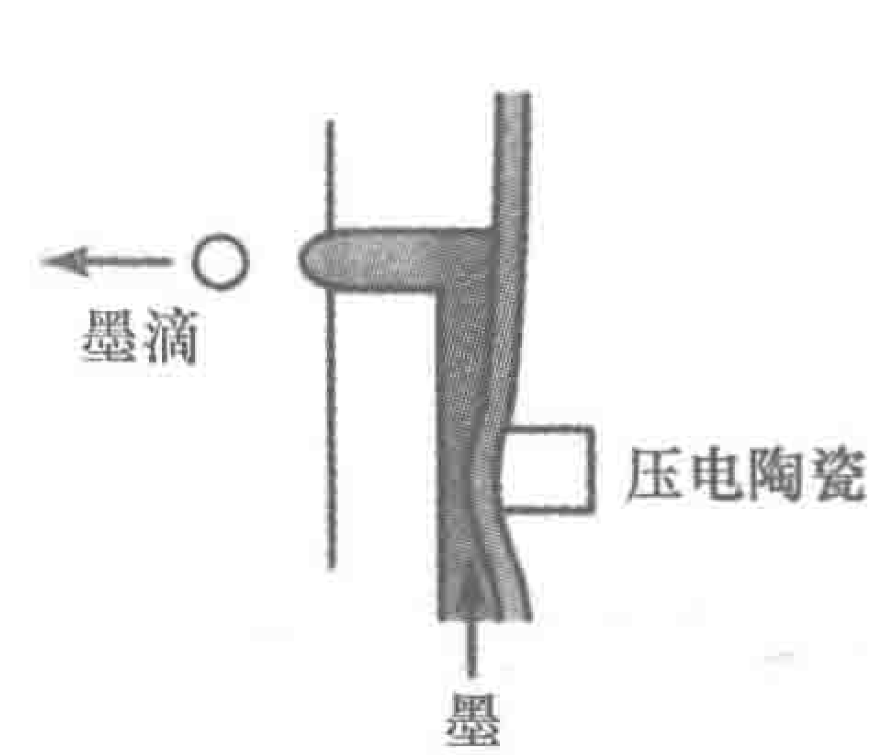


图6-7　产生一系列的墨滴

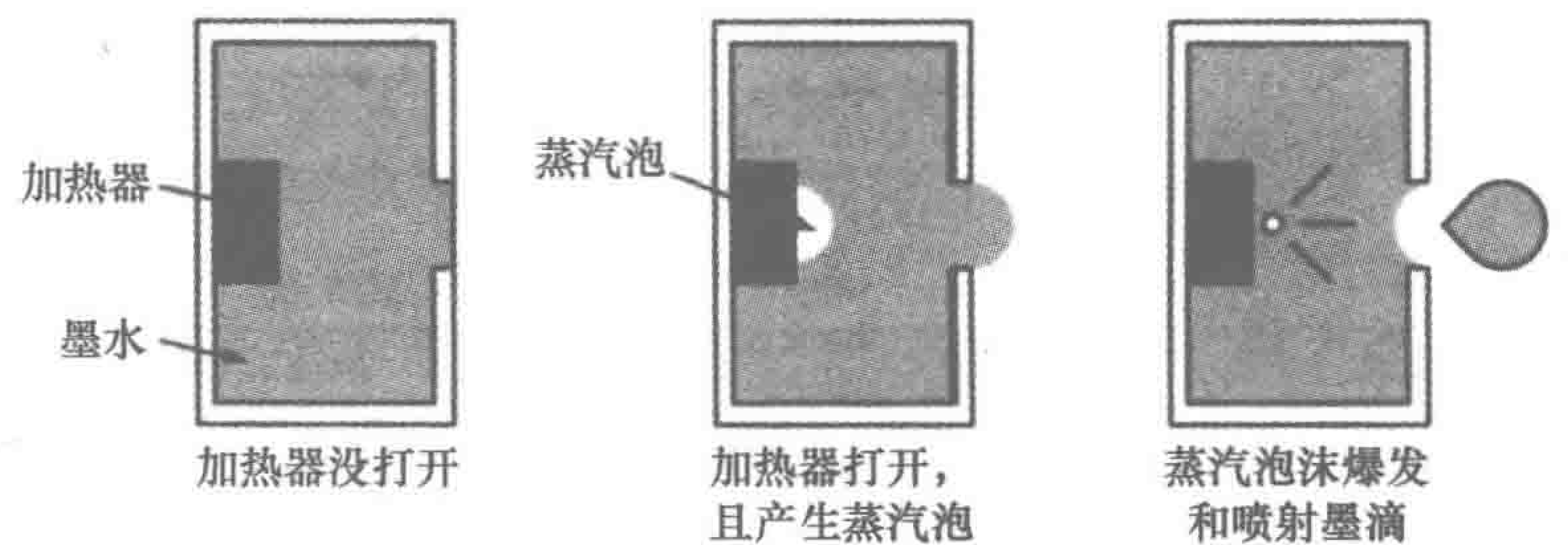


图6-8　气泡喷射原理

激光打印机有感光硒鼓（见图6-9）。在黑暗中，硒有高阻值，当它接近充电线时被充电，该线处于高电压下，当离开该线时电荷泄漏。一个小型八面镜在旋转时可以反射激光光束，并沿着滚筒长度方向扫描。当光线射到硒鼓上时，它的电阻就会下降，并不再带电。通过控制光束的亮度，可以控制滚筒上的点放电或者保持带电。当硒鼓经过色剂存储器时，硒鼓的充电区域就吸附碳粉到硒鼓上了。当纸张通过另一个充电线（即所谓的电晕线）时被充电，于是在纸张接近硒鼓时，就把硒鼓上的碳粉吸附下来。热定影辊是用来高温加热碳粉颗粒的，因此当纸张经过两个辊之间后，碳粉颗粒就被紧紧地贴在纸张上面。目前，通用激光打印机能够每英寸打印1200个点（即1200DPI）。

141

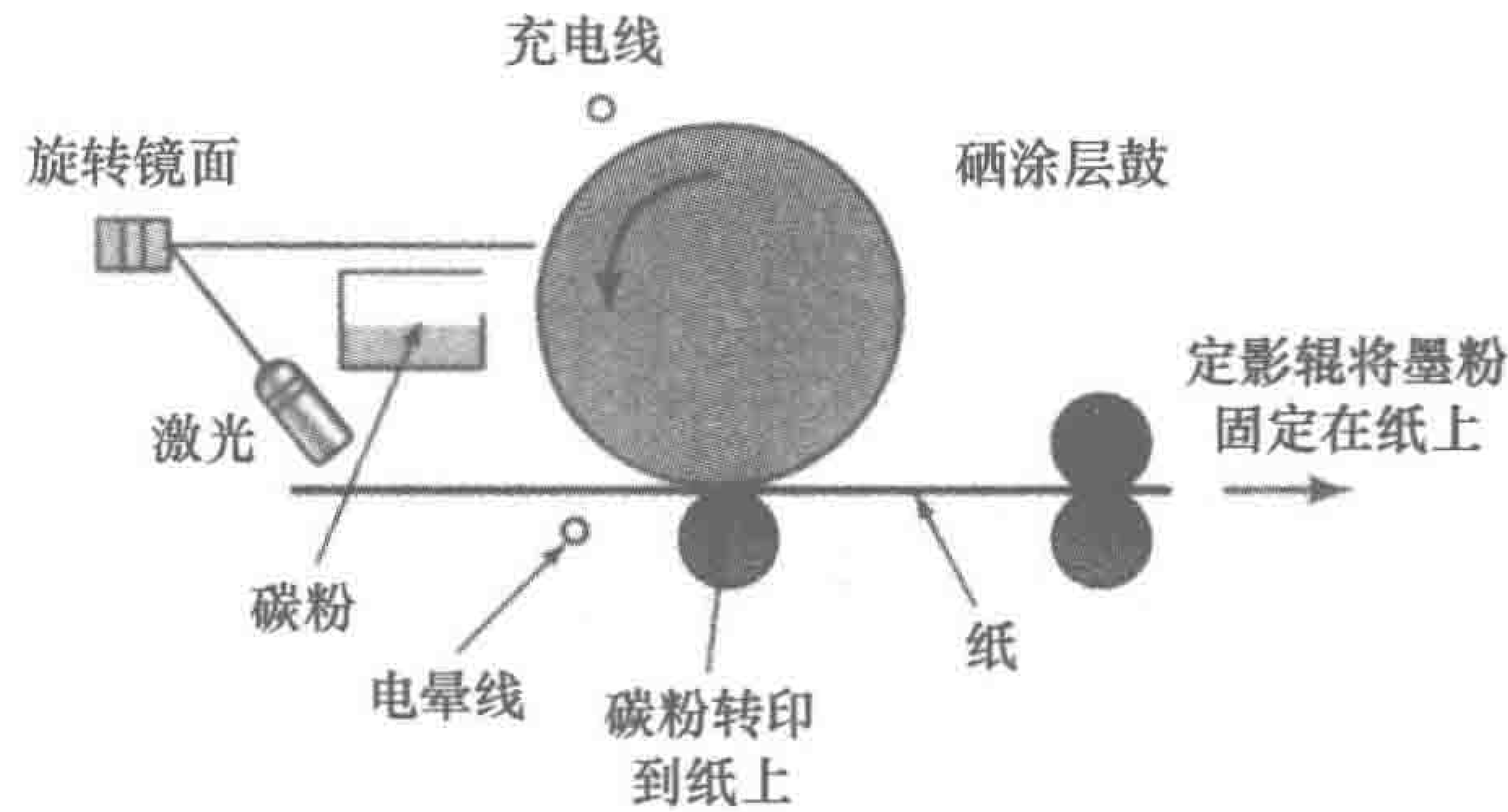


图6-9　激光打印机的基本元素

6.3　磁记录

磁记录是用来在计算机的软盘和硬盘上存储数据的方法。其基本原理是：一个可以响应输入信号的记录头，在薄薄的一层磁性材料上产生相应的磁模式；还有一个读头，可以把磁性材料上的磁模式输出，转换为电信号。此外，还需要一个运动系统，以便于可控制磁材料在头下面运动。

图6-10a给出了记录头的基本组成元素，它包括一个铁磁材料的铁心，上面有气隙。当磁化电流通过缠绕铁心的线圈时，就产生了磁通量。磁性塑料涂层和气隙很近，这意味着磁通量很容易通过铁心和气隙下的磁心涂层。当磁通量穿过磁性涂层时，就被磁化了，这样输入的电信号就被记录下来。改变电流方向可改变磁通量的方向。

重放磁头（见图6-10b）和记录头的结构类似。当磁化涂层接近气隙时，磁通量就在铁心里产生了。铁心中磁通量的变化在缠绕铁心的线圈中感应出了感应电动势。因此，线圈的输出

是一个电信号，而且是与涂层的磁记录数据相关。 142

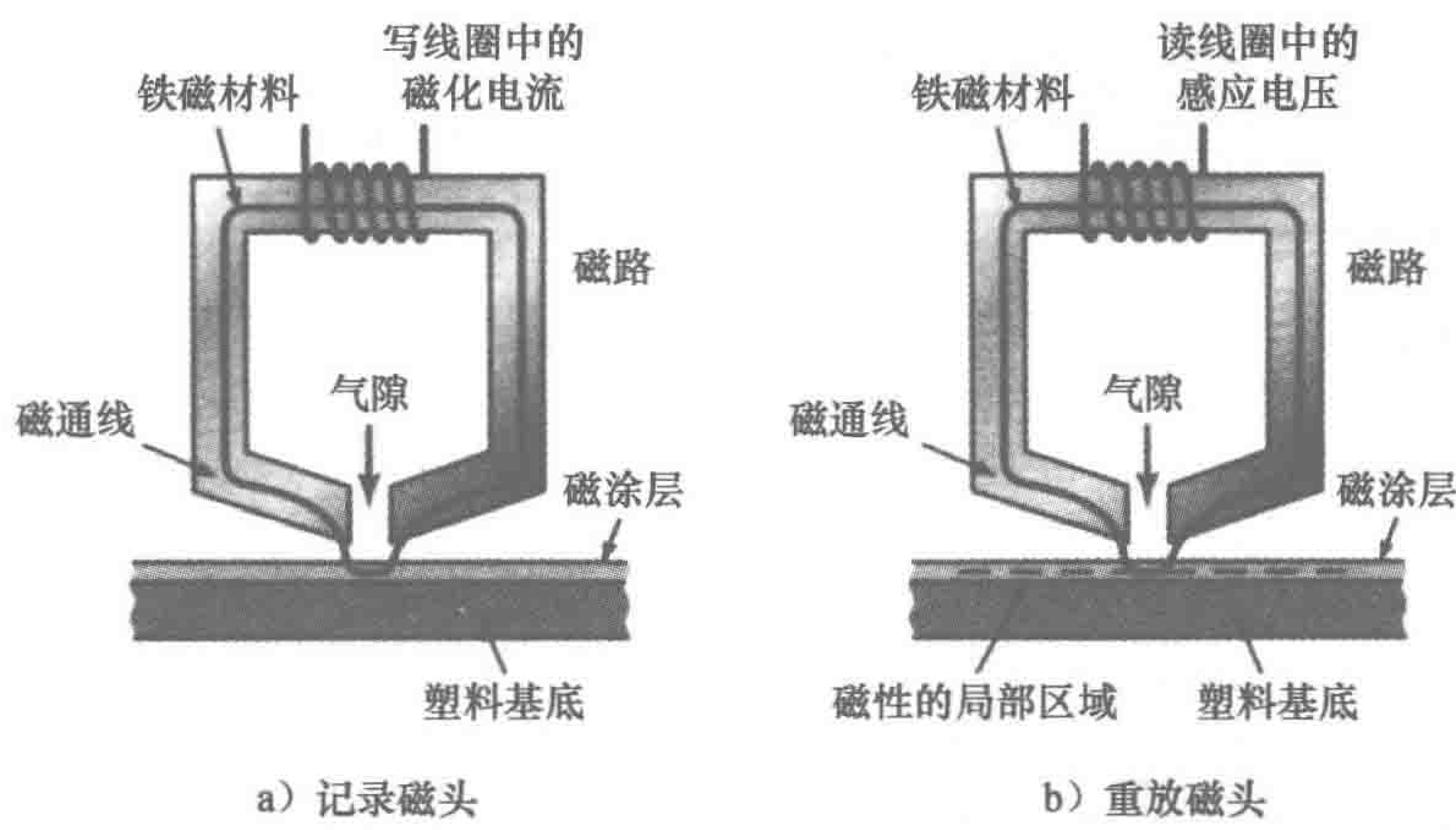


图 6-10 磁基本组成

6.3.1 磁记录码

数字记录是把信号编码成位的组合。1 位存储单元是磁性涂层上的一个元素，其磁性要么是在一个方向上完全饱和，要么是在另一方向上完全饱和。饱和的含义是指当磁性材料达到了最大的磁通量时，继续增加磁化电流磁通量也不会变化。

磁性表面上的位存储单元可能是如图 6-11 所示的形式。把数据记录到磁材料上的磁通量在一个方向表示 0，在相反方向表示 1。然而，我们在读取每一个位元时，需要准确地记录对应的时间，以便于能够知道在什么时间读到哪个位元。然而问题出现了：如果一个外部时钟用于提供时间信号，那么当时间信号和磁性表面移动速度之间出现很小的不匹配时，就有可能漏掉一个位元，或被重复读取。因此，同步是至关重要的。这种同步方法是通过使用位元本身产生的信号来进行采样。一种方法是利用磁场由一个方向上的饱和变成另一个方向上的饱和的这段过渡，即当两位之间的划分界限很明显时，给时钟发生器一个反馈信号来调整它，这样它就与位元同步了。

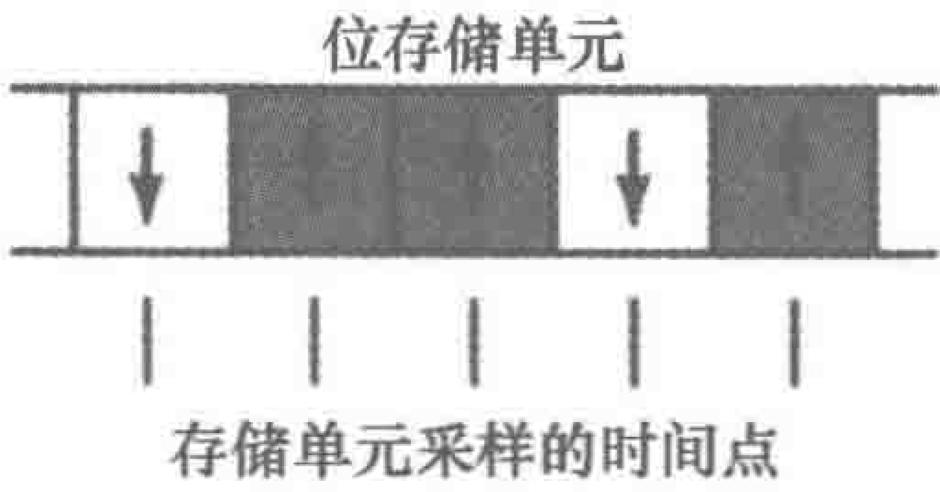


图 6-11 位存储单元

如果磁通量反转发生得不够频繁，这种同步方法仍然可以导致同步上的错误。解决这个问题的方法之一是使用一种特殊形式的编码。常用的一些方法概述如下。

1. 不归零编码（NRZ）

在这种系统中，将磁通量记录在磁带上，当没有磁通量变化时是 0，当有变化时是 1（见图 6-12a）。该方法仍然不是自己来提供时钟的。

2. 相位编码（PE）

相位编码的优点是自己可以提供时钟信号，不需要外部时钟。每个位元一分为二，一半是正向饱和磁通量，另一半是反向饱和磁通量。于是，0 可以表示为半个正向饱和磁通量后面跟着半个反向饱和磁通量，1 可以记录为半个反向饱和磁通量后面跟着半个正向饱和磁通量。由正向反的过渡状态代表 0，由反向正的过渡状态代表 1（见图 6-12b）。 143

3. 频率调制（FM）

这是一个自带时钟的方法，和相位编码很相似但是又有些不同：在每个位元的开始总会有一个磁通量的转换（见图 6-12c）。代表 0 的位元中没有磁通量的反转，但是代表 1 的位元却有磁通量的反转。

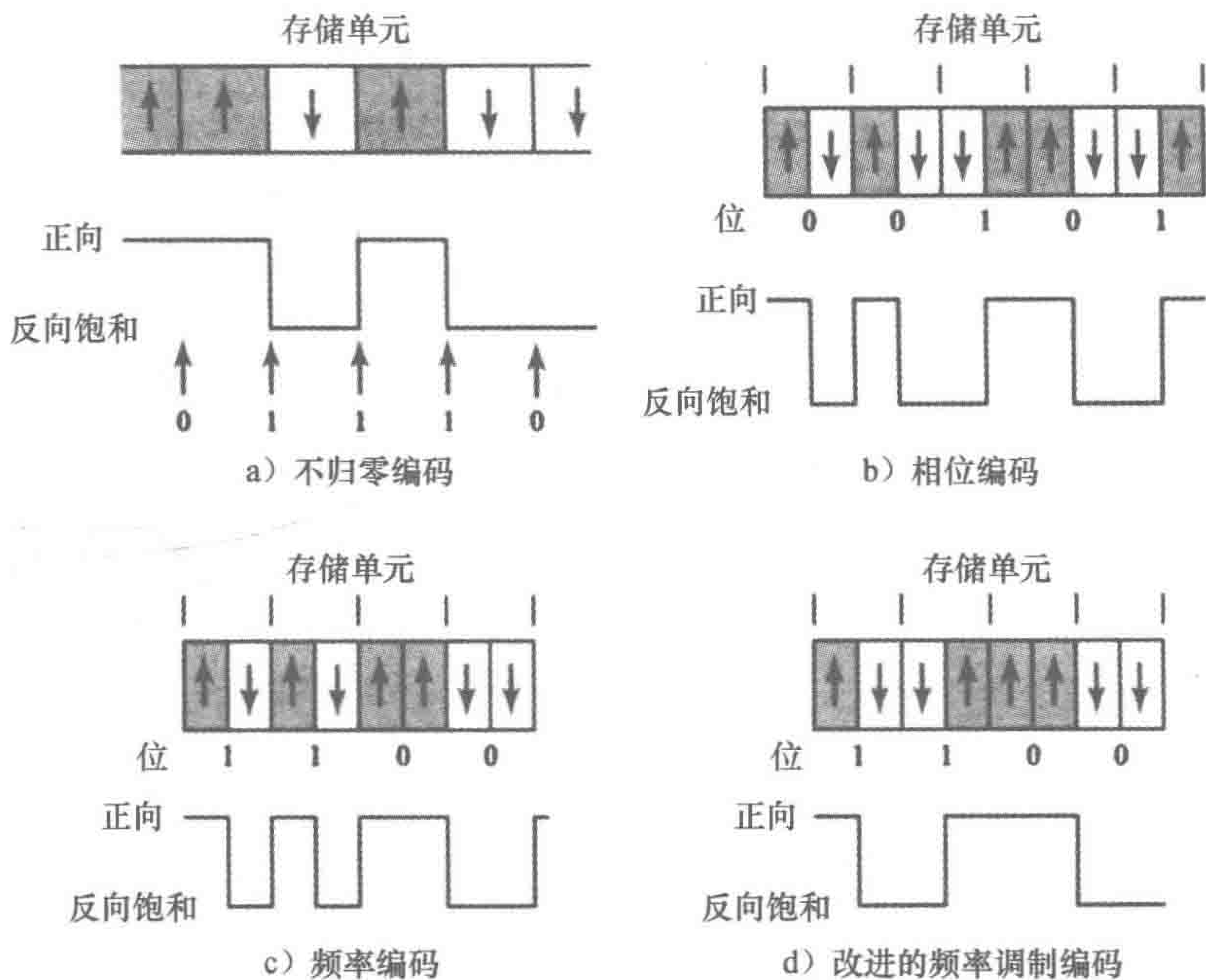


图 6-12　编码方法

4. 改进的频率调制（MFM）编码

这是频率调制码的一种改进（见图6-12d），只有当这一位或者前一位是0的时候，这种差别才能显现出来。这意味着，每个位元只需要一次磁通量的反转。它和游程长度受限代码（RLL）都是通常用于磁盘的方法。

5. 游程长度受限（RLL）编码

这是一组自带时钟编码，它指定了一个最小和最大距离，在磁通量反转之间运行。为了编码能自带时钟，最大游程要足够短，以保证磁通量反转得足够频繁。一个常用的形式是 $RLL_{2,7}$，2 和 7 表明磁通量反转之间的最小距离是 2 位，最大距离是 7 位。编码序列描述为一个 S 代码和 R 代码。S 代码（空间代码）没有磁通量的反转，但是 R 代码（反转代码）有磁通量的反转。2 个 S/R 代码用来表示每位。2、3 或 4 位为一组，每组指定的代码序列为：

144

位序列	代码序列	位序列	代码序列
10	SRSS	011	SSRSSS
11	RSSS	0010	SSRSSRSS
000	SSSRSS	0011	SSSSRSSS
010	RSSRSS		

图 6-13 显示的编码可以实现序列 0110010，它被分成 011 和 0010 两组，所以可以表示为 SSRSSSSSRSSRSS。R 代码之间至少有 2 个 S 代码，但不能多于 7 个。

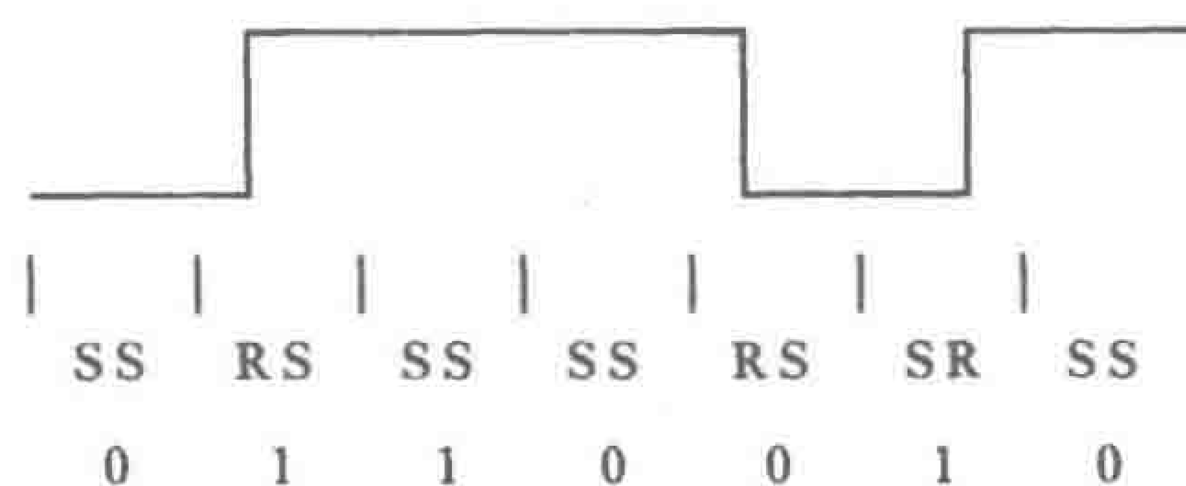


图 6-13　RLL 编码

最优码使得各位可以尽可能近地编码打包在一起，而且在读取的时候不会出错。读数头可以很容易地定位反转，不过它们不能离得太近。RLL 码具备更加紧凑的优势，PE 和 FM 占据了大部分空间，MFM

和 NRZ 占用相同大小的空间。然而 NRZ 却无法自己来提供时钟，这是相比其他编码的一个缺点。

6.3.2　磁盘

对硬盘来说，数字记录是很频繁的。数字数据存储在磁盘表面的磁道上，磁盘的磁道是一个个同心圆，一个磁盘有很多这样的磁道。读/写头用于在磁盘表面运动，通过机械驱动器正/反旋转来访问不同的磁道。磁盘在驱动下旋转，读/写头把数据写入某个磁道。硬盘是把数据存储在沿同心圆的磁盘表面的密封单元（见图 6-14a）。硬盘中装配有多个这样的磁盘，数据存储在磁盘两侧的磁层中。磁盘高速旋转，读/写头通过移动访问磁盘磁道。大量数据可以存储在这种组装在一起的磁盘中。现在存储量达到数百 GB 是很常见的。

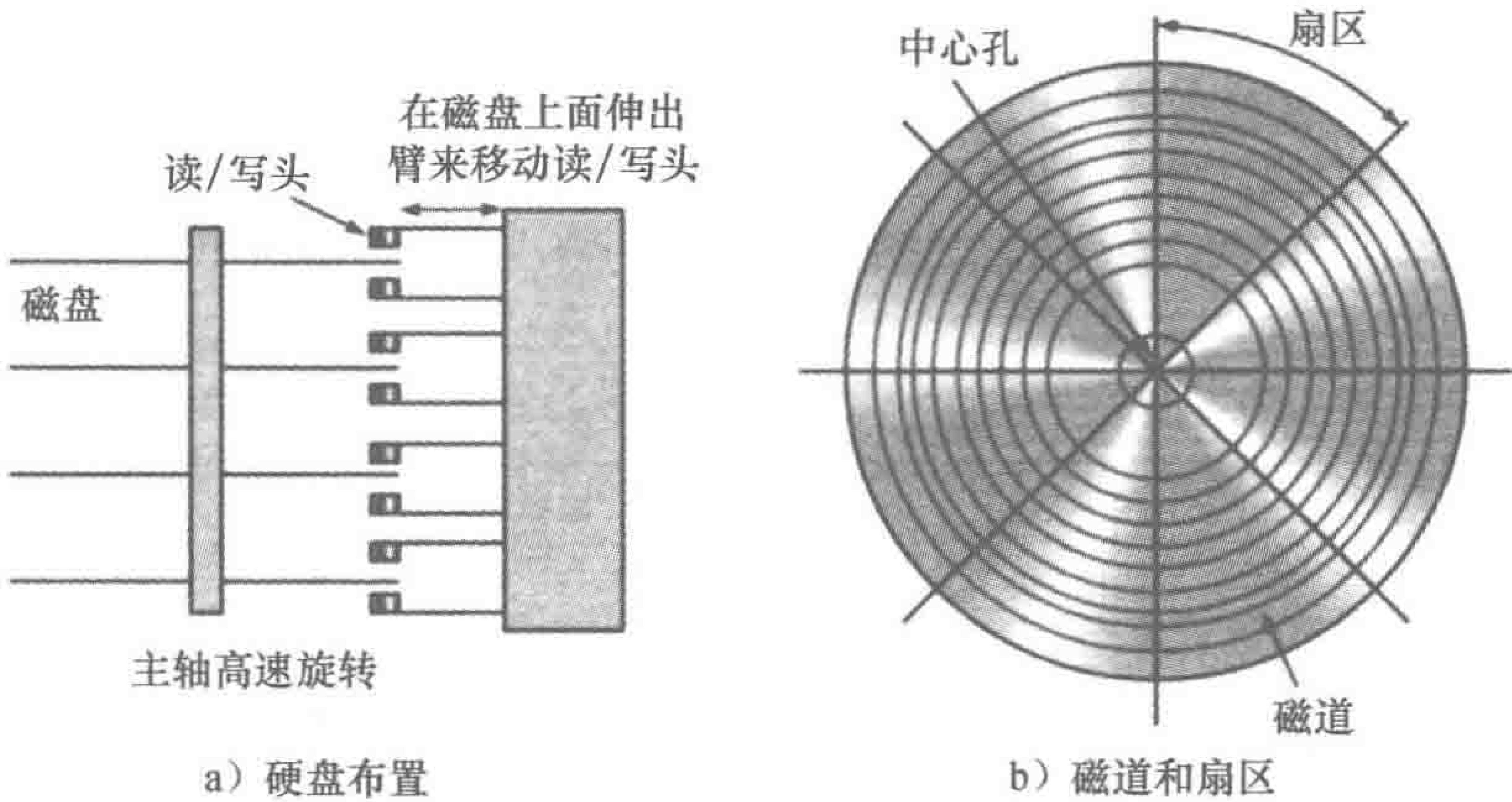


图 6-14　硬盘

145

磁盘表面分为扇区（见图 6-14b），所以信息的地址包括扇区号和磁道号两部分。软盘通常是 8～18 个扇区和大约 100 个磁道。硬盘可能大约有 2000 个磁道和 32 个扇区。为了寻找数据，读/写头必须在磁道上移动，这个过程所花的时间称为**寻道时间**，然后等待，直到需要的扇区移动到下面，这段时间称为**等待时间**。为了确定一个地址，需要标志扇区和磁道的信息。写入这个信息的过程称为格式化，在磁盘存储数据之前必须要进行格式化。这种技术经常用于在磁道上存储地址信息，从而当保存数据时，磁道上的信息序列变为：

```
索引标记,
Sector 0 header,sector 0 data,sector 0 trailer,
Sector 1 header,sector 1 data,sector 1 trailer,
Sector 2 header,sector 2 data,sector 2 trailer,
等等。
```

索引标记包含确定扇区头的磁道号。扇区尾包含循环冗余检查等信息，它可以用来检查一个扇区是不是被正确读取。

6.4　光学记录

和磁盘一样，CD-ROM 是沿着光道存储数据的。不像磁盘有一系列的同心磁道，CD-ROM 有一个螺旋光道。光盘表面涂有高反射铝层。通过激光器发出的直径约为 1μm 的光束在光盘表面进行聚焦雕刻，将信息存储在一系列 0.6μm 宽的光道上。而照射的光束是否会被反射，则取决于它们是被射在了凹坑里还是射在了平整的区域。读取的数据就是反射与非反射的一系列光脉冲。

光记录和磁记录采用类似的编码方法，通常采用 RLL 编码方式。由于光学记录很容易被表面的划痕损坏，或由于灰尘而阻碍激光束阅读数据，所以必须采取方法来检测和纠错。一种方法是**奇偶校验**。采用这种方法时，在一组数据位中还额外增加了一位，这一位或者设置为 1，或者设置为 0，而相应地这组数据中为 1 的位数要么是奇数，要么是偶数。读信息的时候，如果某一位损坏了，那么位的个数就会改变，于是就检测到了错误。

146

6.5　显示器

很多显示系统使用灯光指示器来指示开/关状态或者显示字母数字。术语 alphanumeric（字母数字）是 alphabetic（字母）和 numeric（数字）的缩写，包括所有的字母和数字 0 ~ 9 以及小数点。这样形式的显示器包括七段"灯"，来生成字母和数字。图 6-15 显示了各段，表 6-2 显示了如何用一个 4 位二进制码输入来点亮各个段。

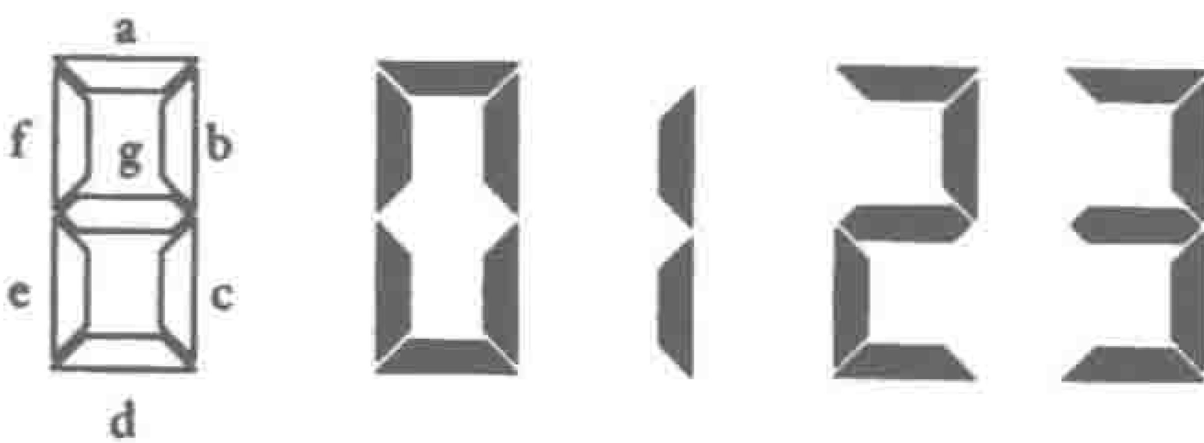


图 6-15　七段显示器

表 6-2　七段显示器

二进制输入				被激活的段							被显示的数字
				a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9

另一种格式包括 7 × 5 或 9 × 7 的点阵（见图 6-16），通过点亮适当的点来生成字符。

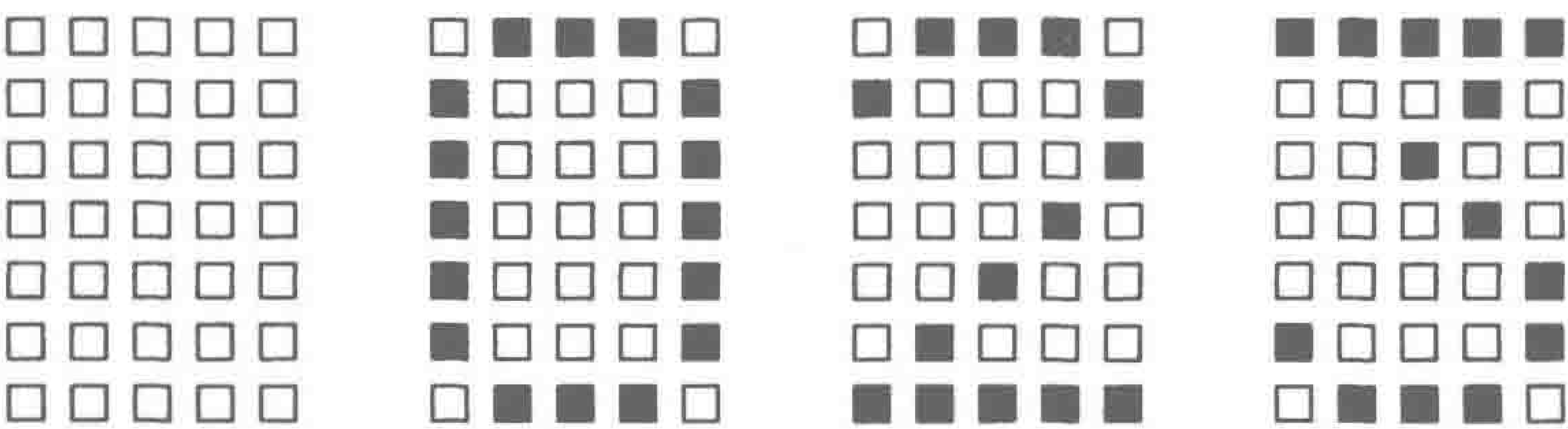


图 6-16　一个 7 × 5 点阵显示器

这类显示器中的光指示器可能是氖气灯、白炽灯、**发光二极管**（LED）或**液晶显示器**（LCD）。**氖气灯**需要高电压和低电流，可以直接使用电网电压供电，但只能用于发出红光。**白炽灯**可以在很宽的电压范围下使用，但需要较高的电流。它们发出白光，所以可以使用镜头来产生任何所需的颜色。其主要优点是亮度很高。

147

6.5.1 发光二极管

发光二极管仅仅需要低电压和低电流，并且很便宜。这些二极管正向偏置时发出某一个波长的光。图 6-17 给出了发光二极管的基本形式，它发出的光通过反射在一个方向得到增强。常用的发光材料是砷化镓、磷化镓和二者的合金。最常用的发光二极管呈红色、黄色或绿色。在基于微处理器的系统中，发光二极管是最常用的指示器。

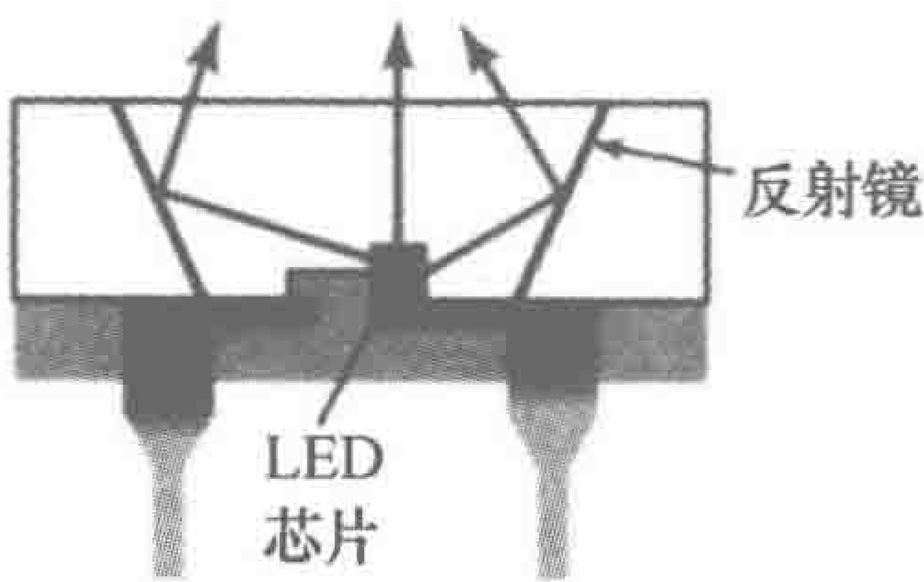


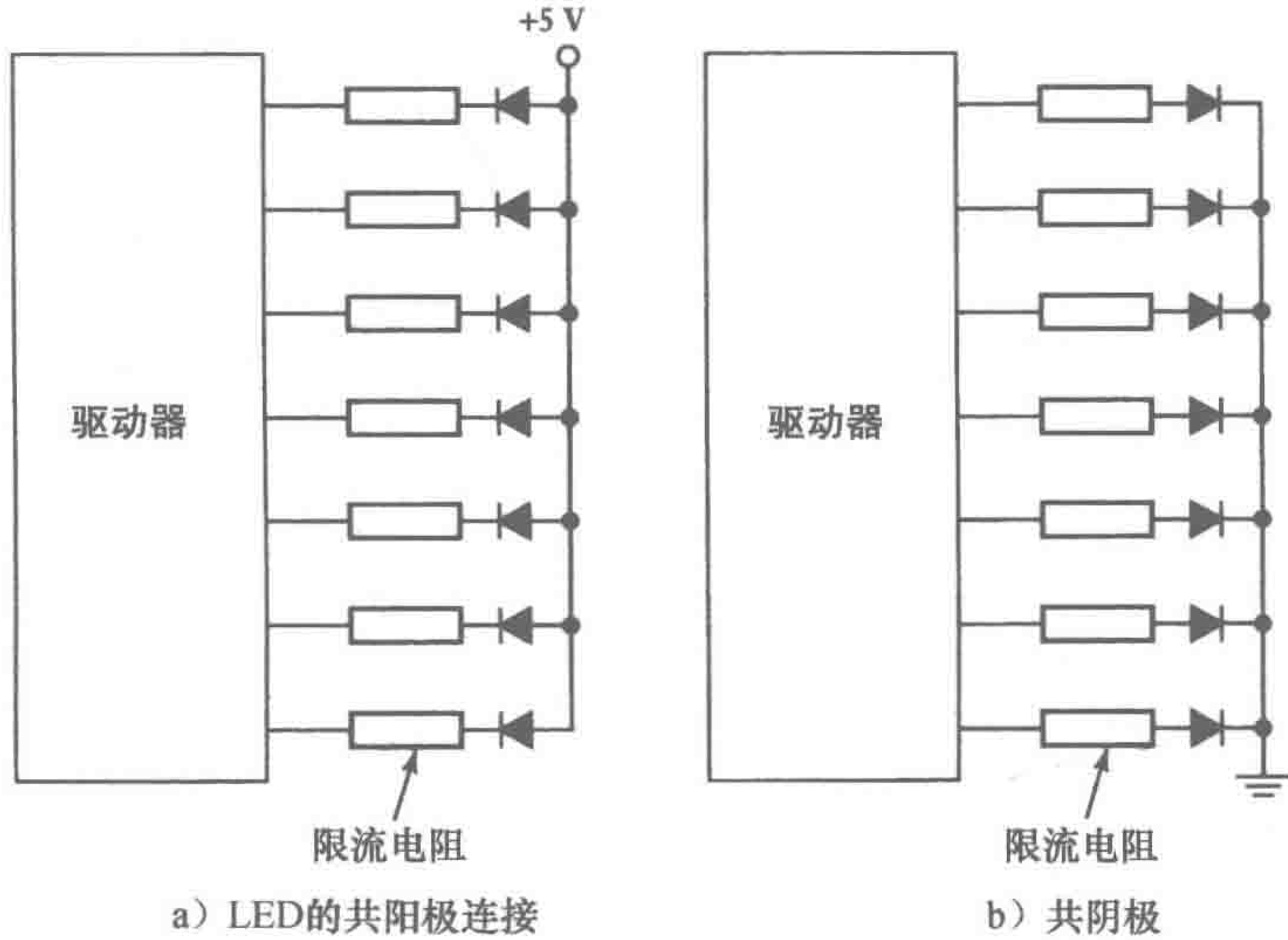
图 6-17 LED

在 LED 的使用中经常需要限流电阻，从而可以把电流限制在 10～30mA 的额定范围内。典型的二极管在 2.1V 的电压下一般会流过 20mA 的电流。也就是说，对于 5V 的供电电压，需要使用分压电阻分去 2.9V 电压。这意味着需要一个阻值为 2.9/0.020 = 145Ω 的电阻，所以通常使用一个 150Ω 的标准电阻。有些发光二极管提供了内置电阻，这样就可以直接连接到微处理器系统中。

发光二极管可作为单光源显示，也可以用于七或十六段字母数字显示器，还可以用在点阵格式和条形图中。

图 6-18a 中显示了如何用七个发光二极管得到如图 6-16 所示的形式，显示系统可能连接在一个驱动器上，通过驱动某一线为低电位，把电压加到 LED 上，使对应线上的 LED 打开。在 LED 发光前，电压必须大于"开启"值。典型的开启电压约为 1.5V。由于所有发光二极管阳极连接在一起，所以这样的装置称为连接的**共阳极**形式。另一种安排是**共阴极**（见图 6-18b）。在共阳极连接方式下，如果要发光二极管发光，就要把电平设置为低电平；在共阴极下，则要设置为高电平。共阳极是常见的选择，这是由于此时电流的方向和大小通常是最合适的。

148



a）LED的共阳极连接 b）共阴极

图 6-18 发光二极管的连接例子

这种类型显示器的例子是 7.6mm 和 10.9mm Hewlett Packard 高亮度七段显示器，共阳极或共阴极形式都很常用。二极管除了可以显示字符外，也可以显示小数点。通过点亮不同的段，可以形成所有的数字和部分字母。

通常驱动器的输出不是常见的二进制形式，而是 **BCD 码**（见附录 B）。在 BCD 码中，每个十进制数字分别以二进制形式编码。例如，十进制数 15 中，1 的编码为 0001，5 的编码是 0101，因此 15 的 BCD 码是 00010101。驱动器的输出然后会被解码成显示器需要的格式，7447 是一个常用于驱动显示器的解码器（见图 6-19）。

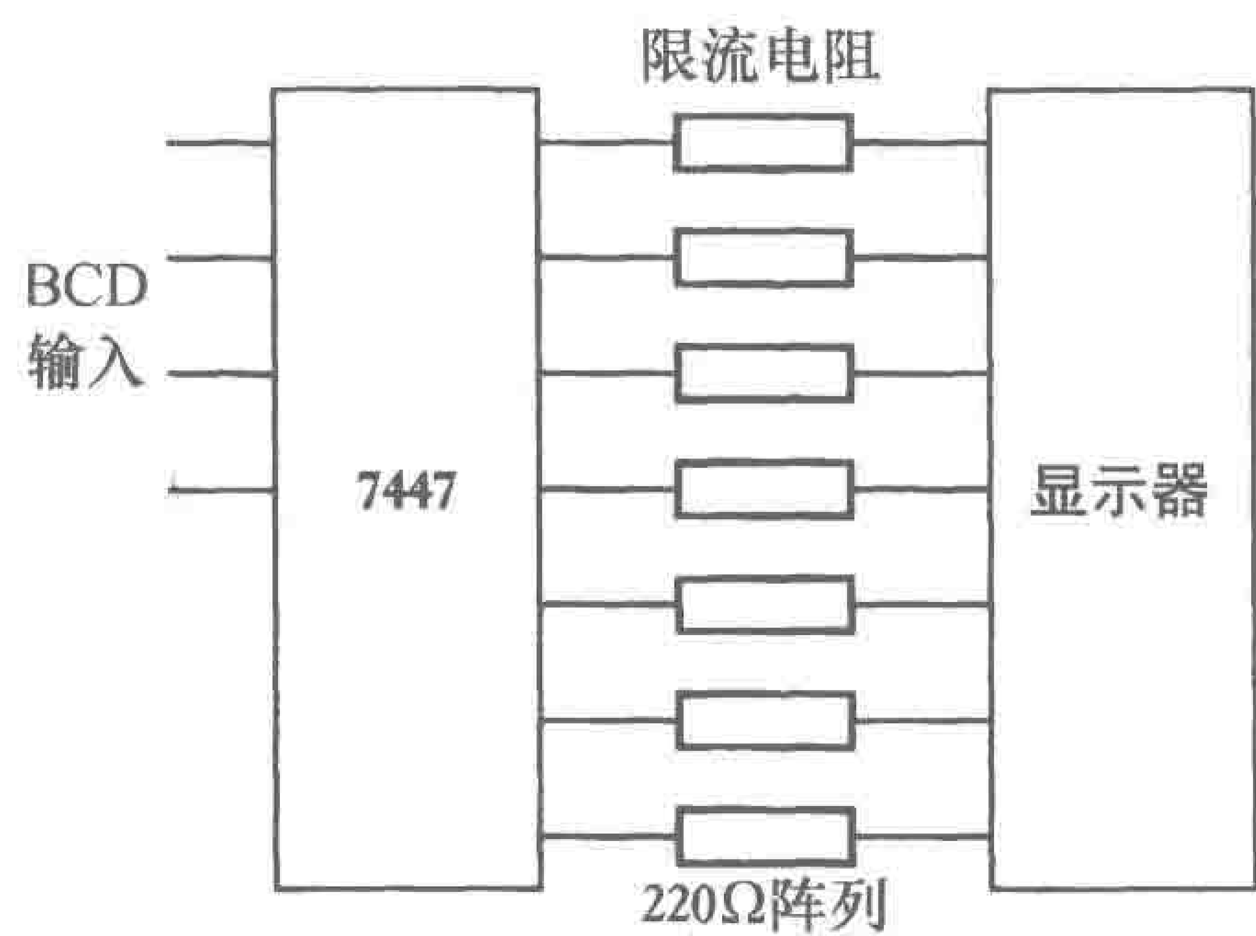


图 6-19 七段显示器的解码器

图 6-20 是用于 5×7 点阵显示器的基本形式。该阵列由 5 列连接器组成，每列连接 7 个发光二极管的阳极。每一行连接到 5 个发光二极管的阴极。要打开一个 LED，只需要将电源加载到它对应的行和列。这种显示方式能显示出所有的 ASCII 码字符。

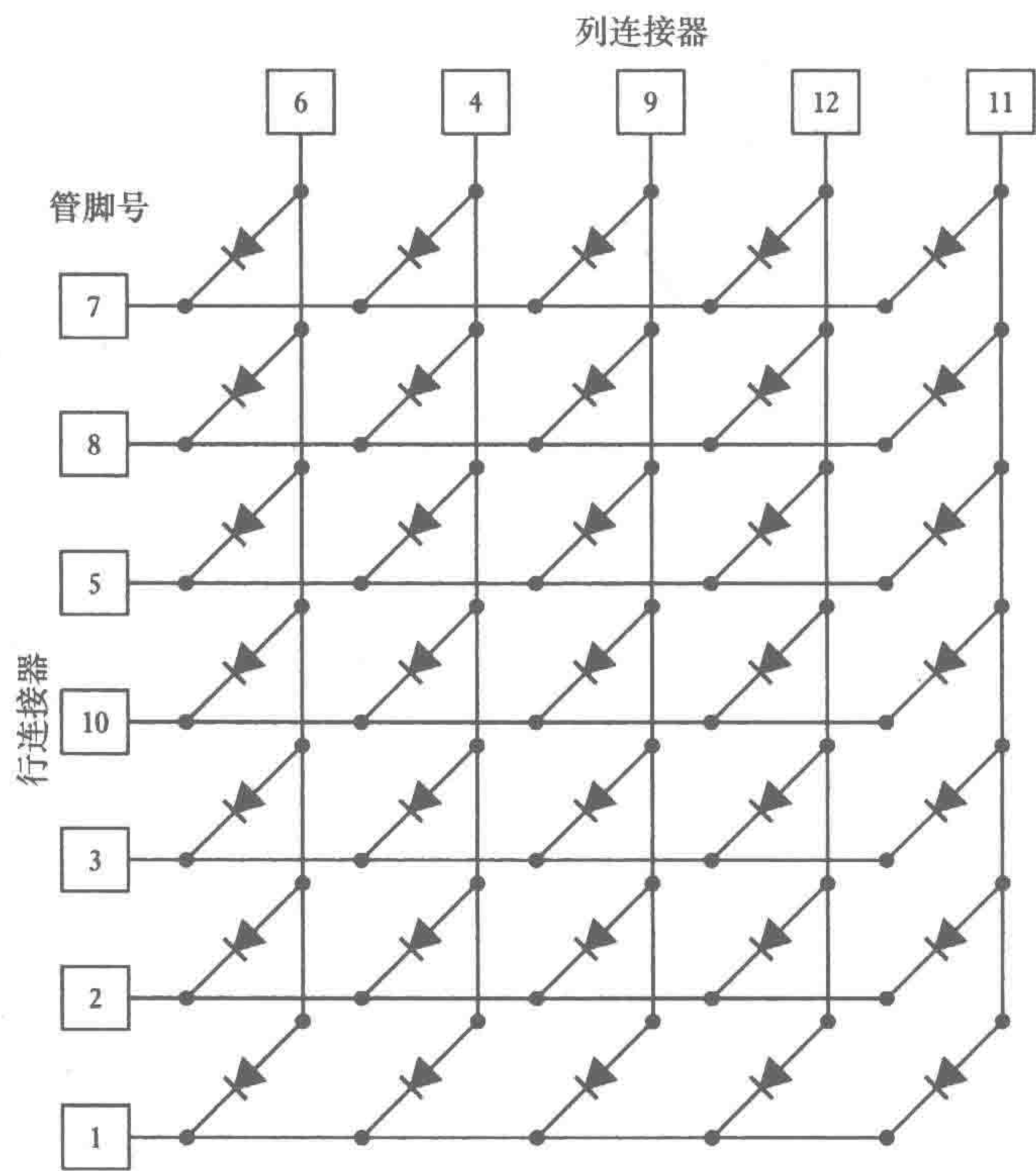


图 6-20 点阵显示器

6.5.2 液晶显示器

液晶显示器不产生任何光，自己依靠反射光或透射光来工作。液晶材料是一种长棒状分子的复合物，它夹在两层有微观槽的聚合物之间。上下两层槽的方向成 90°角。液晶材料的分子与聚合物的凹槽对齐，且在二者之间发生平滑的 90°扭转（见图 6-21）。

当平面偏振光入射到液晶材料时，它的偏振平面会旋转。因此，如果它是夹在 2 片偏振片之间，并且两个偏振片的透射方向成直角，那么旋转使光可以传播，因此材料发出光。

然而，如果电场施加在材料上，分子就会和电场方向一致，通过顶部偏振片的光不会转动，不能通过下边的偏振片而被吸收。于是这种材料就变暗了。

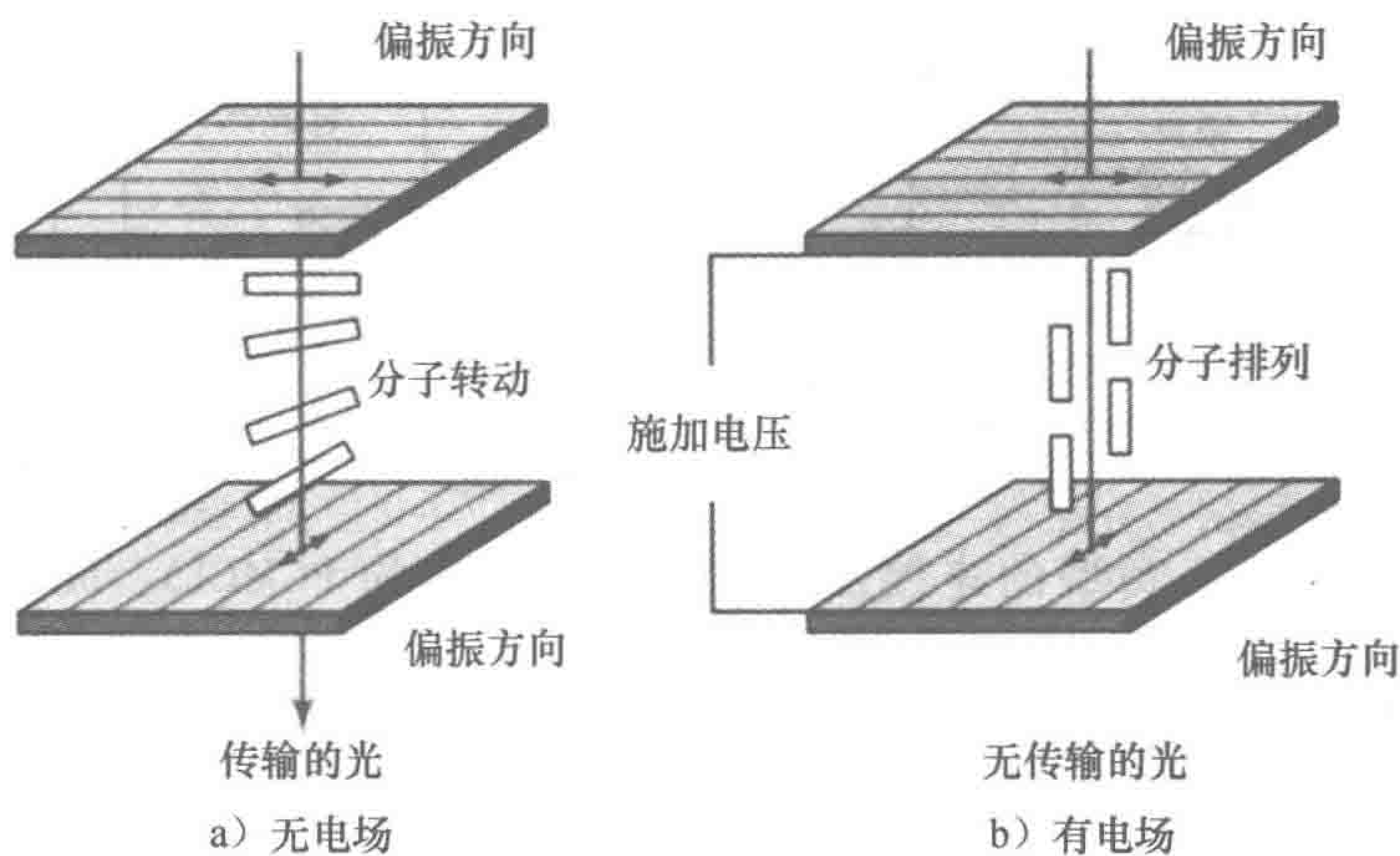


图 6-21　液晶

149~150

在两片玻璃片之间安装透明的电极。发光二极管显示器可以被穿透也可以被反射。对于可穿透的显示，是背光显示。当偏振平面旋转至可透光状态时，显示器就亮了，否则它是暗的。反射光显示，在晶体的背面有一个反射表面。因此，当入射光穿过显示器时，它会反射回来，所以是亮的。当入射光不能通过显示器时，它就是暗的。

液晶显示器可以有很多显示形式，包括与七段 LED 显示器类似的七段显示器。对不同的显示元件施加电压，会使得在液晶显示器的明亮背景上显示出黑色的字迹，背景上没有电场。可以使用一个 3 ~ 12V 的交流电场打开一段显示。驱动电压不能使用直流，而必须使用交流，因为直流电场会产生反应，从而破坏晶体。液晶显示器的响应时间比较长，通常是 100 ~ 150ms，但是功耗非常低。

液晶显示器也可以作为点阵显示器。液晶显示模块还可显示一行或多行字符。例如，可以显示两行 40 个字符。

集成电路驱动器可以驱动 LED。因此，MC14543B 可用于七段 LCD 显示器。当输入的是 BCD 码时，该驱动器是可用的。5 × 8 点阵显示器可以用 MC14500 驱动。例如，日立 LM018L 是 40 字符 ×2 行的反射型液晶显示模块，内置有 HD44780 驱动器，它提供了一系列功能，包括 192 个 5 ×7 点字符和 8 个用户自定义的字符，因此可直接和一个 4 位或 8 位微处理器接口。

LED 显示器经常使用在电池供电的设备上，如手机、手表和计算器。

6.6　数据采集系统

数据采集（DAQ）在从传感器获取数据并转成数字形式被计算机处理、分析和显示的系统中经常用到。该系统包括：传感器、连接线（用于连接传感器和信号调理器，实施滤波和放大）、数据采集硬件（用于把输入信号转化为数字格式或把输出信号转化为用于控制系统的模拟格式）、计算机和数据采集软件。该软件用于分析数字信号。这样的系统也经常用于执行控制功能。

6.6.1　计算机与插件板

151

图 6-22 显示的是带插件板的数据采集系统的基本组成要素，插件板安装在一台用于数据采集的计算机上。输入到插件板之前的信号调理与传感器有关，例如它可能是以下情况。热电偶：放大、冷端温度补偿、线性化；应变片：惠斯通电桥（Wheatstone Bridge），提供电桥电压和线性化；电阻温度检测器（RTD）：提供电流和线性化。

在选择数据采集板时，必须遵循以下原则：

1）应用什么类型的计算机软件系统，如 Windows 还是 MacOS？

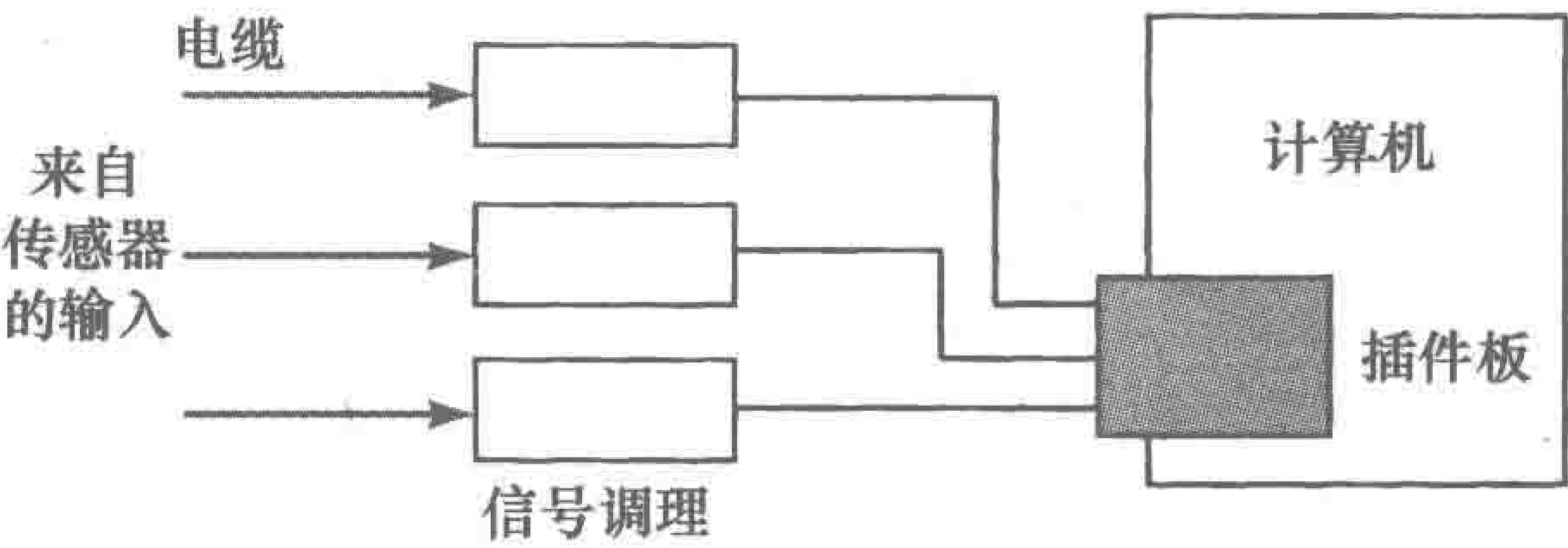


图 6-22　数据采集系统

2）插线板插入什么类型的连接器，如笔记本电脑的 PCMCIA、MacOS 的 NuBus，还是 PCI？

3）需要多少模拟输入？它们的范围是什么？

4）需要多少数字输入？

5）需要怎样的分辨率？

6）需要的最低采样频率是多少？

7）是否需要定时或计数信号？

图 6-23 显示了一个数据采集板的基本要素。有些采集板被设计成只处理模拟输入/输出和其他的数字输入/输出信号。

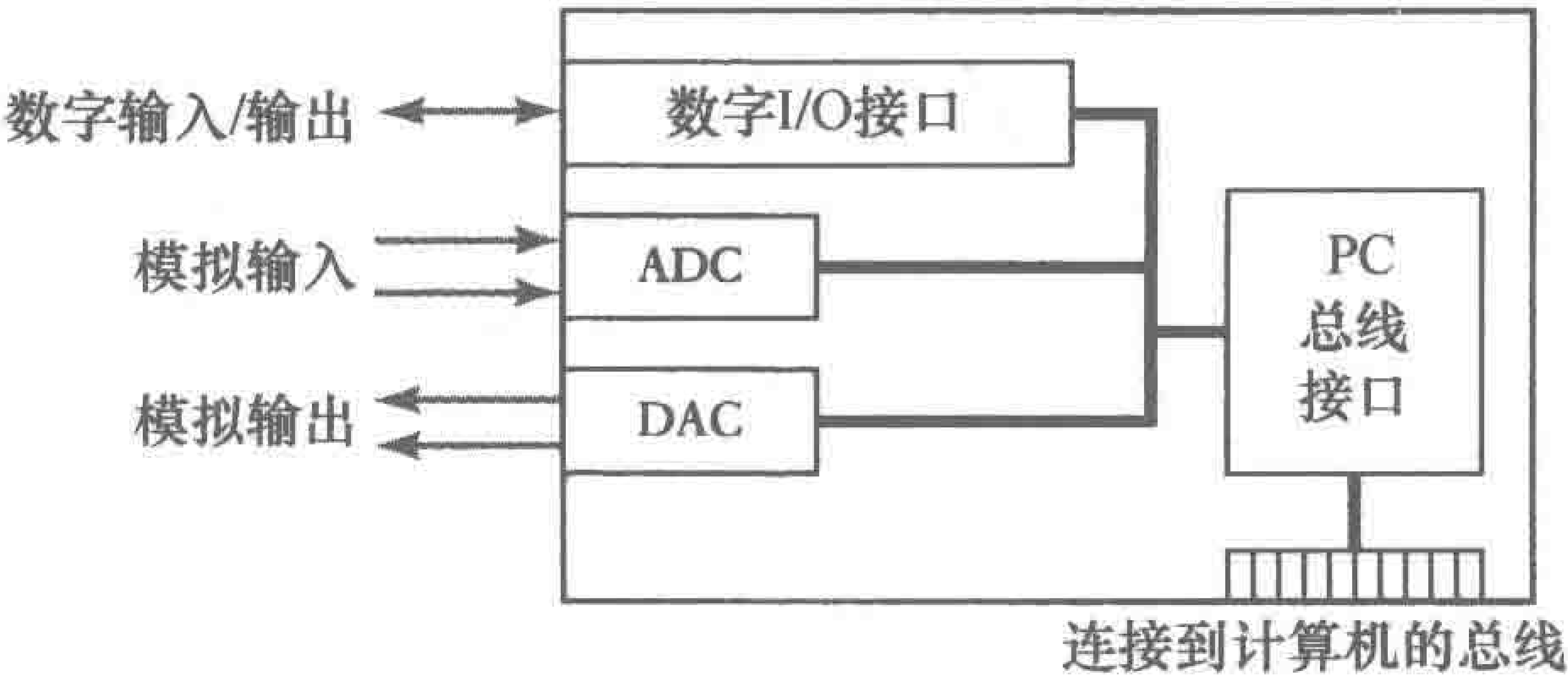


图 6-23　数据采集板的基本要素

所有数据采集板使用的**驱动**、程序软件一般由采集板的制造商提供，它还会与计算机通信，告诉计算机已插入的是什么硬件，计算机也可以和采集板进行交互。使用采集板之前，3 个参数必须设置：输入/输出通道的地址、中断级别和 DMA 通道。应用 Windows 软件使用即插即用插件板时，可以用软件设置参数。此外也可按附带的用户手册在插件板上进行跳线设置。

应用软件可以用来协助设计测量系统和分析数据。LabVIEW 就是这样的一款由 National Instruments公司开发的图形化的编程软件，用于数据采集和仪器控制。LabVIEW 称为**虚拟仪器**，因为在外观与操作上都与实际的仪器类似。虚拟仪器有三个部分：1）有人机交互式界面的仪器面板，上面包含控制旋钮、按键和图形显示器；2）可用于编程的方块图，上面有源代码，可通过计算机屏幕显示的各个图标之间的连线来完成程序的编制；3）如果希望把编好的程序应用于其他的方块图中，还可以把该程序变成一个图标和连接器，提供一个虚拟仪器的图形化表达。

152

图 6-24a 是为虚拟仪器选择的一个图标，其中有一个模拟采样从一个指定的输入通道中输入。“Device” 是分配给 DAQ 板上的设备号，“Channel” 是数据源通道，“Sample” 是一个数模转换，“High limit” 和 “Low limit” 是电压信号的限幅（默认值是 +10V 和 -10V，改变这些值会自动改变放大器的增益）。

如果我们需要从每个通道获得一个波形，就要选择图 6-24b 的图标。对于每一个输入通道，都要在一段时间内以指定的采样速率获得一系列的采样，并给出模拟量的时变输出波形。

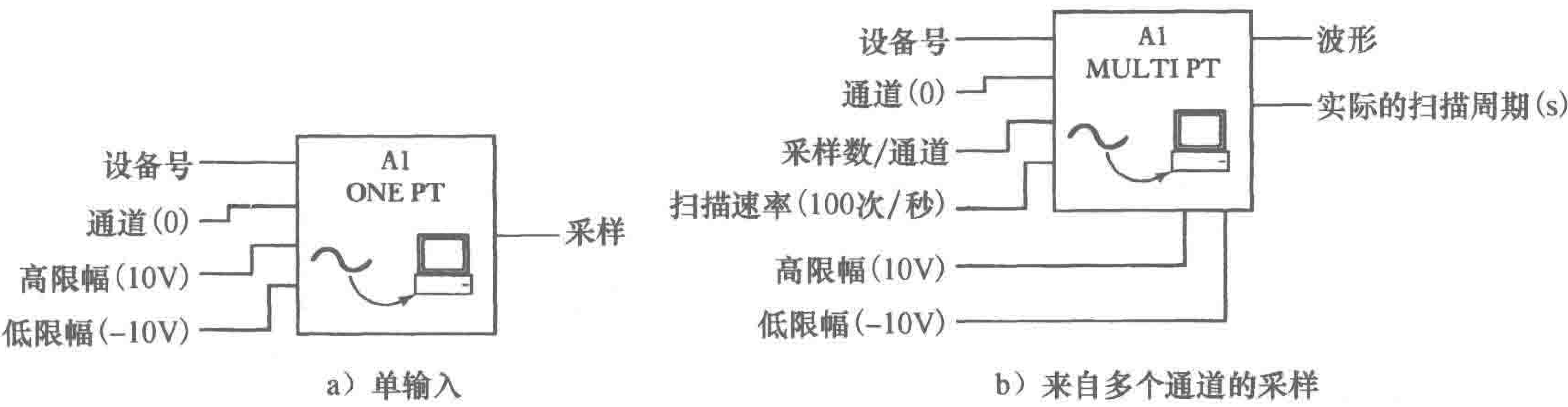


图 6-24　模拟输入图标

通过连接其他的图标，可以建立一个方框图，这可能需要使用大量的模拟通道，顺序采样后，把结果显示为一系列的图像。图 6-25 给出了一种可能的面板类型，用于简单的 DAQ 采集。通过使用向上和向下箭头可以改变参数，并显示出来。

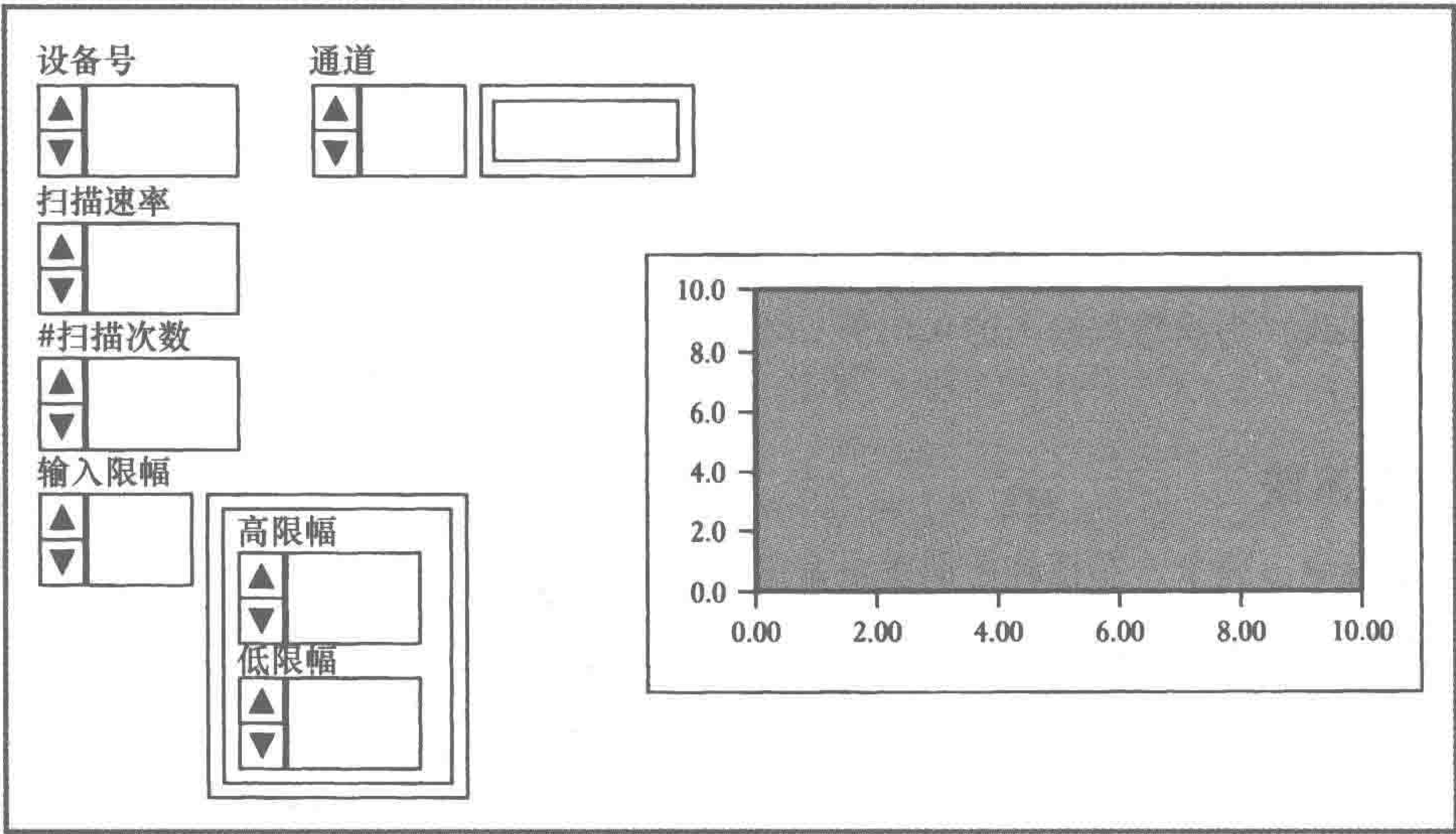


图 6-25　面板

相比于传统仪器，虚拟仪器有一个很大的优势。传统仪器的性质和接口已经被供应商确定且无法改变，而虚拟仪器都可以由用户定义且容易改变。

153

6.6.2　数据记录器

数据记录器用于 DAQ 系统且能独立于计算机使用。计算机将程序设定好，再把该程序存入记录器的存储卡中，或者直接从计算机将程序下载到记录器上，就可以使记录器具有 DAQ 功能。

图 6-26 给出了一个数据记录器的基本组成。这样的数据记录单元能监控来自很多传感器的输入。来自不同传感器的输入经过信号调理后，传送给多路复用器，多路复用器选择传送进来的一个信号，信号经过放大后传送给模数转换器（ADC）。然后由微处理器处理产生数字信号。微处理器能够进行简单的算术运算，例如对测量值求平均。系统产生的输出将显示在一个能同时显示输出量和通道数的数字表上，也可以打印、存盘或者提交给计算机做进一步的分析处理。

数据记录器经常和热电偶一起使用，因此会有特殊的热电偶输入来提供冷端补偿和线性化。多路复用器可轮流切换到每一个传感器上，因此输出是一个样本序列。通过编写微处理器程序能够将多路复用器切换到一个通道上，这样就能获得一个单一源的连续扫描数据或者所有

通道的周期性扫描，例如1分钟、5分钟、15分钟或者30分钟、60分钟的周期。

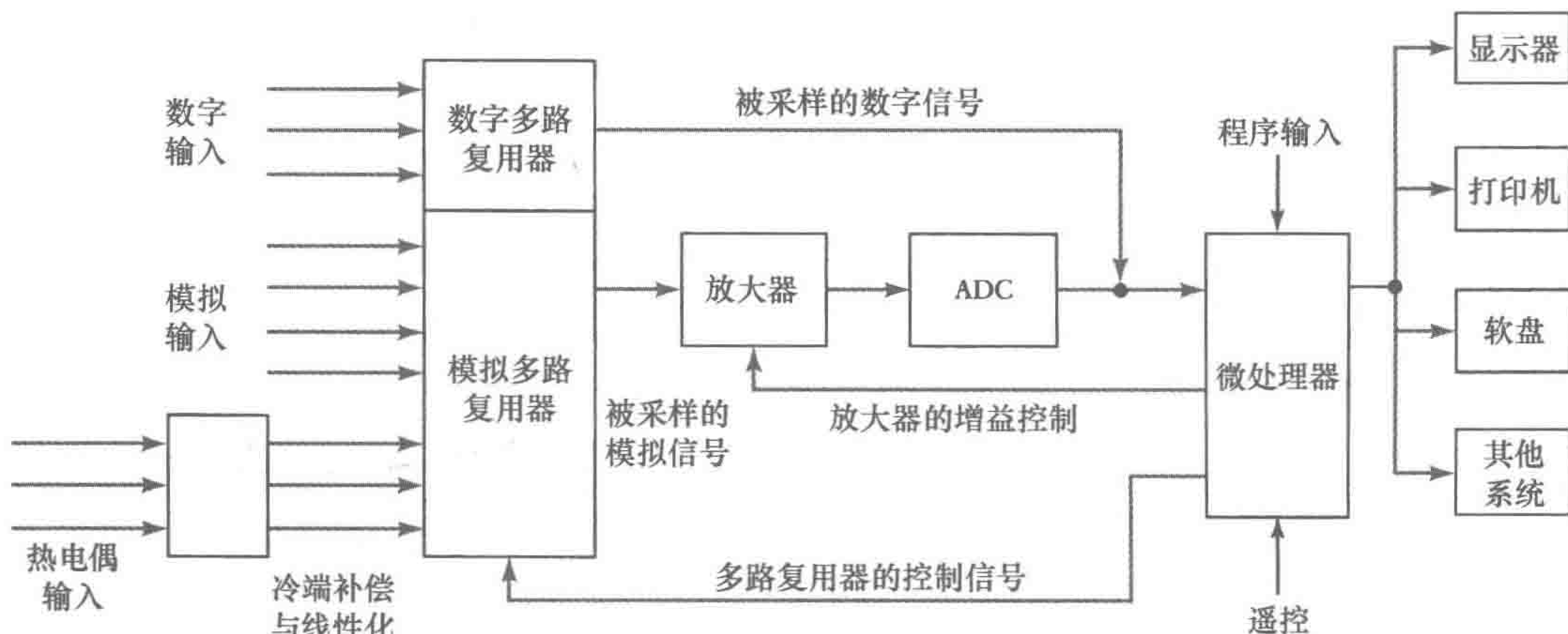


图6-26　数据记录器系统

尽管某些数据记录器能够处理相当多通道的数据（如1000个），但一个典型的数据记录器仅仅能够处理20~100个通道的输入。它们的采样和转化时间约为10μs，能够进行1000次/秒的读操作。精度一般为满量程的0.01%，线性度一般为满量程的±0.005%。在任意输入上的**串扰**（cross-talk）为满量程输入的0.01%，串扰用于描述当传感器采样时其他通道传感器的信号干扰。

154

6.7　测量系统

下面的例子说明了一些特殊应用测量系统的设计要点。

6.7.1　用于检测提起重量的称重传感器

如图6-27所示，一个连杆型称重传感器具有4个附着在连杆表面的应变片，能够放置在重物和提拉重物的缆绳之间。其中两个应变片处在纵向方向上，另外两个处在横向方向上。当连杆受到张力时，纵向应变片会被绷紧，横向应变片会被压缩。假设称重传感器的灵敏度是当连杆施加500MPa的压力时，输出是30mV。假设应变片的应变系数是2.0，电阻值是100Ω。

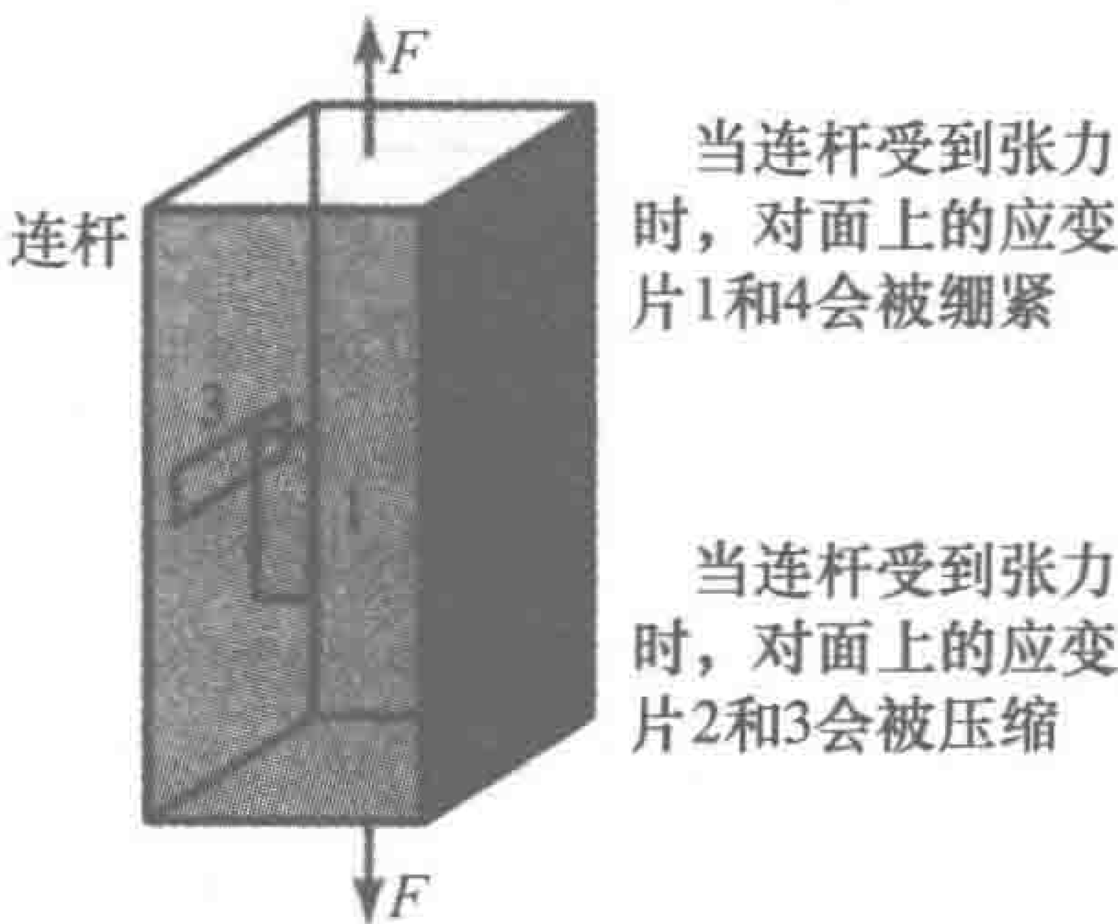


图6-27　称重传感器

当连杆上施加的负载是F的时候，由于弹性模量E是应力/应变，而应力定义为单位面积所承受的作用力，则纵轴方向上的应变ε_1是F/AE，横向应变ε_t是$-vF/AE$，A是横截面积，v是连杆材料的泊松比。应变片（见2.3.2节）对此应变的响应是：

$$\frac{\delta R_1}{R_1}=\frac{\delta R_4}{R_4}=G\varepsilon_1=\frac{GF}{AE}$$

$$\frac{\delta R_3}{R_3}=\frac{\delta R_2}{R_2}=G\varepsilon_t=-\frac{vGF}{AE}$$

惠斯通电桥（见3.5节）的输出是：

$$V_o=\frac{V_sR_1R_4}{(R_1+R_2)(R_3+R_4)}\left(\frac{\delta R_1}{R_1}-\frac{\delta R_2}{R_2}-\frac{\delta R_3}{R_3}+\frac{\delta R_4}{R_4}\right)$$

对于$R_1=R_2=R_3=R_4=R$，$\delta R_1=\delta R_4$，$\delta R_2=\delta R_3$，则：

$$V_o = \frac{V_s}{2R}(\delta R_1 - \delta R_2) = \frac{V_s GF}{2AE}(1 + v)$$

155

假设连杆材料是钢，查表可得 E 约为 210GPa，v 约为 0.3。因此，对于应变系数为 2.0 的应变片，500MPa 应力（$=F/A$）产生的输出为：

$$V_o = 3.09 \times 10^{-3} V_s$$

对于 10V 的供电电压，电桥将输出 30.9mV 的电压。如果这是仅有的需要的负载值，就不需要再经过放大了。可是如果这是一个最大值，而我们希望检测这个值以下的负载，那么我们就需要一个差分放大器。输出可以显示在一个高输入电阻的电压表上（高输入电阻是为了避免负载效应问题）。因此，数字电压表就很合适。

6.7.2　温度警报系统

这是一个测量系统，要求当液体温度超过 40℃ 的时候就触发报警装置。液体温度通常为 30℃，系统输出电压为 1V 信号来操作报警。

因为输出用电信号，所以就要求一个合理的响应速度，一种可能是使用电阻元件。为了从电阻元件产生电压输出，可以使用惠斯通电桥。当液体温度在 30℃ ~40℃ 之间变化时，输出电压可能小于 1V，但差分放大器能够获得要求的电压。可用比较器将真实值和报警设定值进行比较。

假设需要镍（nickel）元件，它的电阻温度系数是 0.0067/K。因此，如果电阻在 0℃ 时的阻值是 100Ω，那么在 30℃ 时的阻值是：

$$R_{30} = R_0(1 + \alpha t) = 100(1 + 0.0067 \times 30) = 120.1\Omega$$

40℃ 时的阻值是：

$$R_{40} = 100(1 + 0.0067 \times 40) = 126.8\Omega$$

因此有一个 6.7Ω 的阻值变化。如果用这种元件制作惠斯通电桥的一个臂，且在 30℃ 的时候达到平衡，那么输出电压是（详见 3.5 节）：

$$\delta V_o = \frac{V_s \delta R_1}{R_1 + R_2}$$

电桥在 30℃ 时平衡，那么所有臂的值是相同的，且供电电压是 4V，则：

$$\delta V_o = \frac{4 \times 6.7}{126.8 + 120.1} = 0.109\text{V}$$

我们可以使用差分放大器把这个电压放大到 1V（详见 3.2.5 节）：

$$V_o = \frac{R_2}{R_1}(V_2 - V_1)$$

$$1 = \frac{R_2}{R_1} \times 0.109$$

156

因此 $R_2/R_1 = 9.17$，所以如果我们使用 1kΩ 的输入电阻，那么反馈电阻必须是 9.17kΩ。

6.7.3　滑轮的角位置

电位计可用于检测滑轮的角位置。假设电位计旋转的最大角度为 320°，请考虑每转 1 度输出 10mV 所需的组件。

当供电电压 V_s 连接在电位计两端时，我们为了保护电位计不会通过过大的电流，要与电位计 R_p 串联一个电阻 R_s。这时电位计两端总的电压降就是 $V_s R_p/(R_s + R_p)$。对于一个满刻度为 θ_F 的电位计，旋转角为 θ 则可输出的电压为：

$$V_\theta = \frac{\theta}{\theta_F} \frac{V_s R_p}{R_s + R_p}$$

假设电位计的电阻是4kΩ，且 R_s 为2kΩ，则每旋转1度输出10mV，我们有：

$$0.01 = \frac{1}{320}\frac{4V_s}{4+2}$$

因此需要供电电压4.8V。为了避免因记录器的电阻而引起负载效应现象，可以使用一个电压跟随器，因此电路应如图6-28所示。

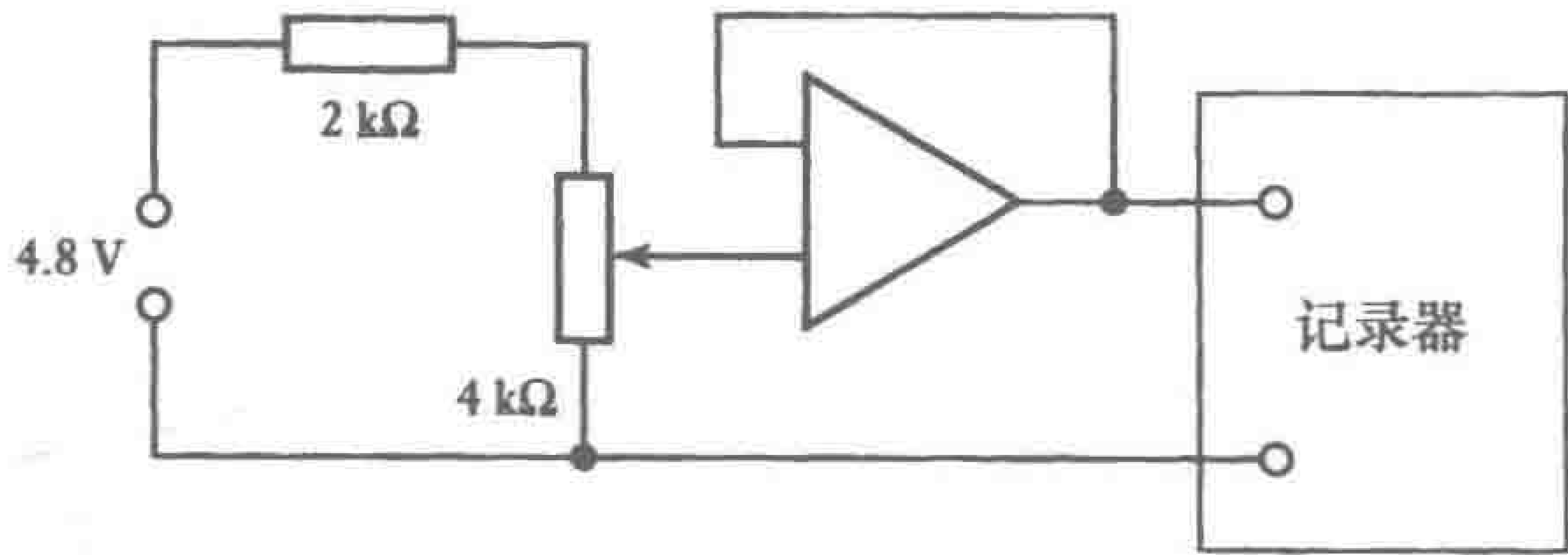


图6-28　滑轮监控器

6.7.4 温度测量的二进制输出

假设一个温度测量系统的温度测量范围在0℃～100℃之间，要求产生一个8位二进制输出，使每1位的变化对应于1℃的变化。该输出被输入到微处理器，作为温度控制系统的一部分。

需要一个线性温度传感器，可以使用热敏晶体管LM35（见2.9.4节）。LM35在供给5V的电压后输出为10mV/℃。如果将LM35的输出连接到一个8位模数转换器，那么可以获得一个数字输出。我们需要ADC的分辨率为10mV，则每10mV的阶跃变化就可以用1位的变化来表示。假设我们使用一个逐次逼近的模数转换器，如ADC0801，这就要求输入一个参考电压，能够分成 $2^8=256$ 位，每一位代表10mv。因此需要2.56V的参考电压。这里需要输入到ADC0801的参考输入电压是 $V_{ref}/2$，所以需要一个精确的1.28V输入电压。这种电压可以通过使用电位器获得，电位器两端跨接5V电源，电路中加入电压跟随器，从而可以避免负载效应问题。由于要求5伏电源电压即使有波动，电压也必须稳定在1.28V。为了实现这个要求，可用电压稳压器，例如有一个2.45V的精密基准电压稳压器ZN458/B，其电路如图6-29所示。

157

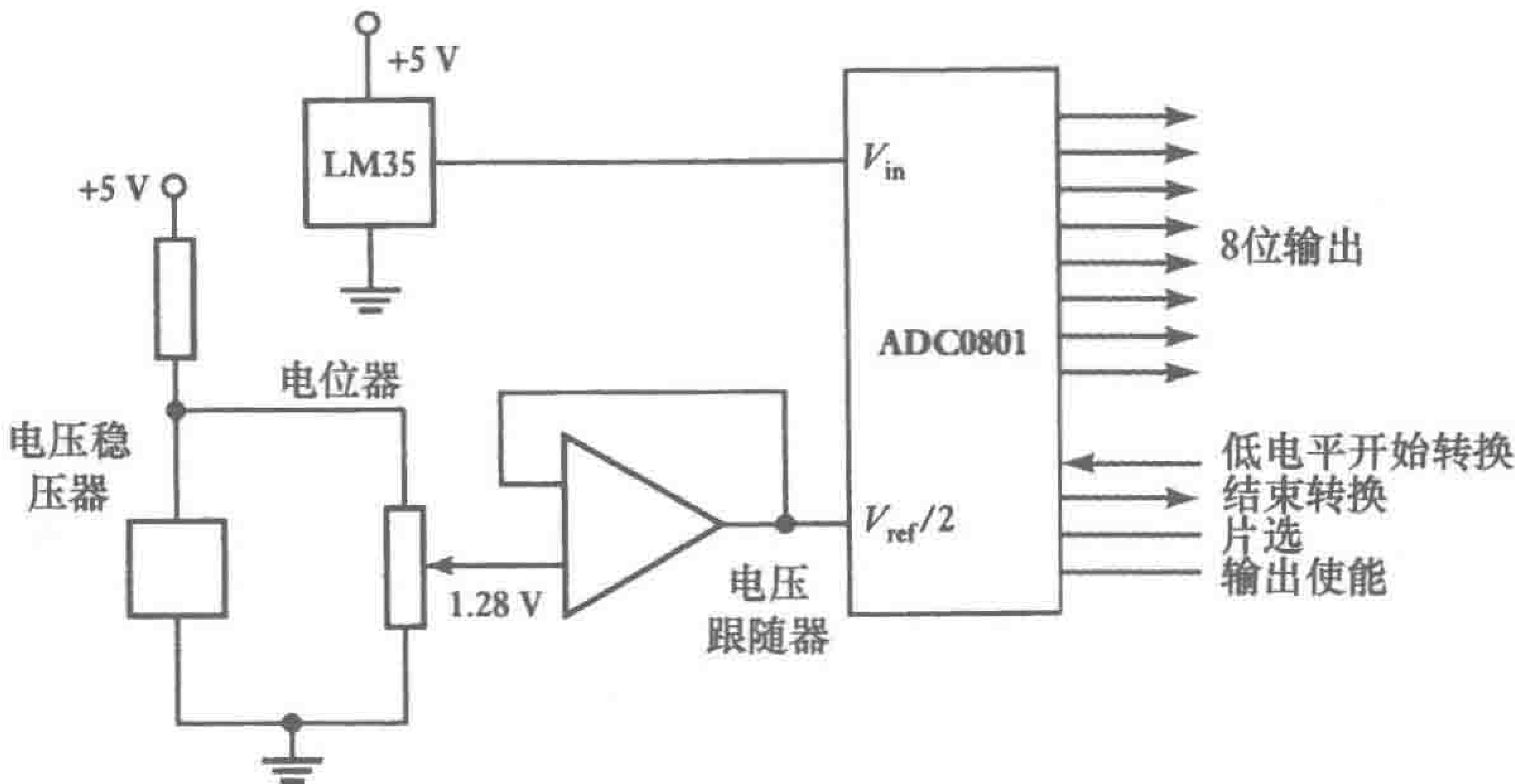


图6-29　温度传感器

6.8 测试和校准

测量系统的测试安装分为三个阶段：

1. 安装前的测试

在安装之前，对每个仪器进行正确校准和操作测试。

2. 管道及布线测试

以气动管线为例，在仪器连接前，要先吹入干净干燥的空气，然后测试压力来确保没有裂缝。对于工艺管道，所有的管道连接前要先经过彻底的冲洗和测试。对于仪表电缆，在与设备连接之前，都应检查它们的连续性和绝缘电阻。

3. 预调试

这部分涉及安装完成后的测试，当连接了控制室内所有的控制面板和显示器后，所有的仪器组件完全都处于操作就绪状态。

6.8.1　校准

校准包括将测量系统及其子系统的输出和已经知道的精度标准进行比较。该标准可能是专门为校准工作而准备的一种仪器，也可能是一些定义标准值的方法。在许多公司里一些标准电阻和器件被保存在一个标准部门，而且只用于校准目的。日常使用的仪器校准和国家标准之间的关系概述如下：

1）国家标准是用于校准那些校准中心的标准。

2）校准中心的标准是用于校准仪器厂商的标准。

158

3）仪器制造商的标准仪器用来提供公司内部的标准。

4）公司内部所用的标准用于校准生产过程的仪器。

有一个简单的可追溯的线索，可以从使用过程的仪器追溯到国家标准。以下是一些可用于公司的校准过程的例子。

1. 电压表

这些可以核对标准电压表或标准元件提供的标准电动势。

2. 电流表

这些可以核对标准电流表。

3. 应变片的应变系数

首先从一批次的应变片中选取一个样本，在安装到某些测试件上之后，向它们施加可测试的应变，这样电阻变化就可以测量出来，于是就知道应变系数了。

4 惠斯通电桥电路

当向惠斯通电桥的一个臂中引入一个标准电阻后，就可以检测惠斯通电桥的输出了。

5. 称重传感器

对于小容量的称重传感器，可以使用标准砝码的负载进行校准。

6. 压力传感器

压力传感器可使用载重（dead-weight）试验器校准（见图 6-30）。通过在活塞托盘中放入标准重量 W 的砝码来产生校准压力。然后，丝杆驱动的柱塞被压入腔体内，从而抬起整个活塞组件的重量。校准压力是 W/A，A 是活塞的横截面积。另外静重试验器可用于校准压力表，然后这个压力表可用于校准其他仪器。

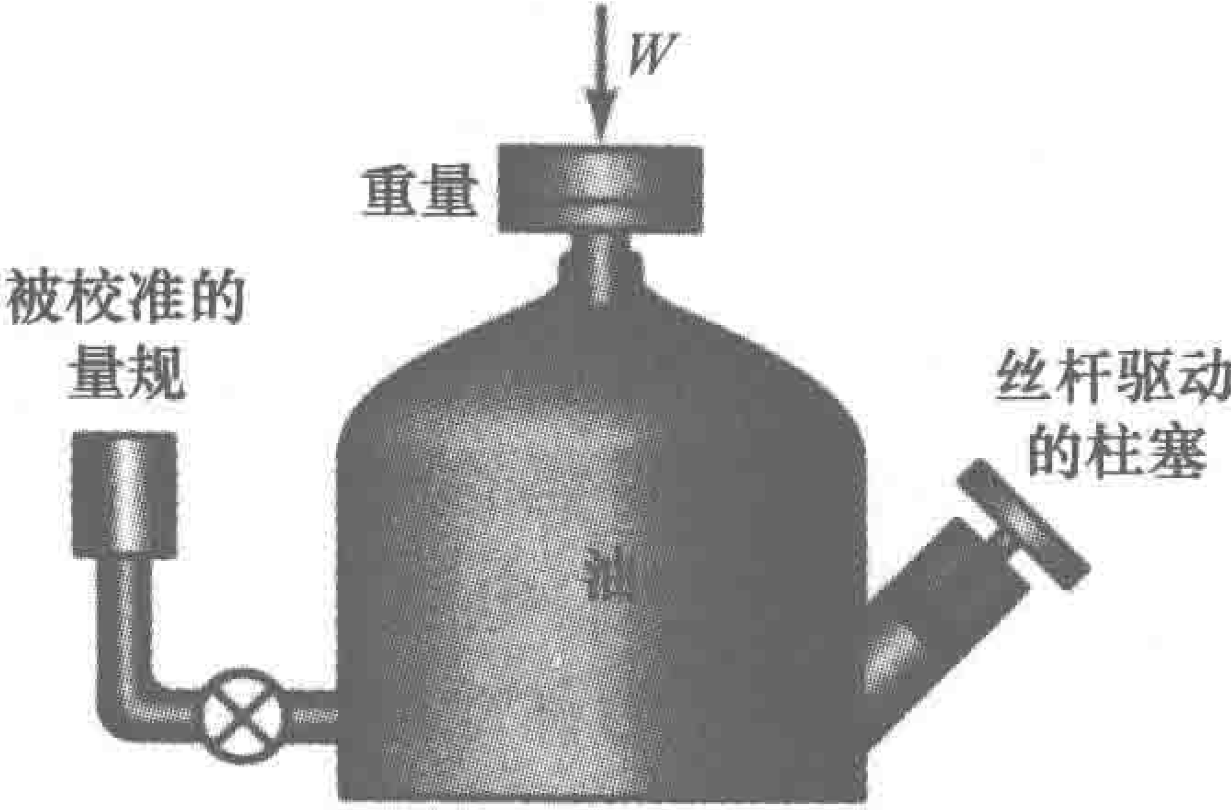


图 6-30　压力传感器的载重校准

7. 温度传感器

可以通过把传感器浸入融化的纯金属或者水里来校准温度传感器。物质的温度是逐渐降低的，于是可以获得一份温度－时间的记录。当物质的

状态从液体变为固体时，温度保持不变。其值都可以从表中查到，因此就可以得到一个用于校准的准确的参考温度。另外，也可以使用液体沸腾时候的温度。然而，沸点取决于大气压，如果所处环境不是标准的大气压，那么一定要做出修正。另外，在公司内部测量系统得出的数值还可以拿来与标准温度计进行比较。

159

本章小结

当将任何测量系统组装到一起时，**负载效应**都是必须重视的一方面，即考虑把新的负载连接在系统任何一个元件的输出终端上时带来的影响。

指示器给出的是一个即时地随着传感器测量变化而变化的可视指示，而**记录器**则是记录一段时间内的输出信号，并自动生成永久的记录。

在从传感器获取数据，并转换成数字形式交给计算机进行处理、分析和显示的系统中，**数据采集**（DAQ）经常会用到。**数据记录器**是可以独立于计算机而使用的 DAQ 系统。

虚拟仪器是由软件生成的工具，在外观和操作上它们都模仿实际的仪器。

检测一套测量系统的安装分为三个阶段：安装前的测试、管道及布线测试和预调试。

校准包括将测量系统及其子系统的输出与已经知道的精度标准进行比较。

习题

6.1 解释下列显示系统指标中术语含义：

(a) 记录器：量程的 ±0.2% 死区（dead band ±0.2% of span）。

(b) 一个硬盘有两张磁盘、4 个读/写头，每个头对应一个磁盘的表面。每个表面有 614 个磁道，每个磁道有 32 个扇区。

(c) 数据记录器：100 个输入，串扰大小为每个输入的满量程的 0.01%。

(d) 双踪示波器：两个相同通道的垂直偏转，带宽是从直流到 15MHz，在 11 个校准步骤中偏转为 10 ~ 20mV/div，在 19 个校准步骤中时间从 0.5μs/div ~ 0.5s/div。

6.2 解释传感器、信号调理器和显示设备构成的测量系统的负载效应问题。

6.3 设计一个显示单元，可以用做：

(a) 热电偶输出的永久记录。

(b) 一套可以显示油压的显示设备。

(c) 一个用来保存微处理器输出数据的记录设备。

160

(d) 当模拟风的湍流加载到飞机上时监测到的瞬时电压。

6.4 如图 2-32 中的一个圆柱形的称重传感器，有 4 个应变片贴在其表面。2 个是在圆周方向，另外两个是在纵轴方向。当这个圆柱受到一个压力负载的时候，轴向应变片将被压缩，而周向应变片会张紧。如果圆柱的横截面积是 A，弹性模量是 E，那么作用于圆柱的力对轴向应变片的应变是 $-F/AE$，对周向应变片的应变是 $+vF/AE$，v 是材料的泊松比。使用称重传感器设计一个完整的测量系统，可用于监测水箱中水的质量。水箱自重为 20kg，在所需要的水位时重 40kg。需要监测的质量的精度在 ±0.5kg 以内。应变片的应变系数是 2.1，电阻都是 120Ω。至于其他的东西，按照你设计的需求来确定。如果你使用低碳钢作为称重传感器的材料，那么拉伸模量可取 210GPa，泊松比取 0.30。

6.5 设计一个完整的测量系统，使用热电偶来确定锅炉中的水的温度，并在仪表上给出指示。温度范围设为 0 ~ 100℃，需要精度达到全量程的 ±1%。确定热电偶的材料和所有其他的必要项目。你的设计必须考虑冷端和非线性问题。你可能会需要参考热电偶表。以下数据是来自冷端在 0℃时的表格，可作为参考：

材料	E. M. F. (mV)				
	20℃	40℃	60℃	80℃	100℃
铜 - 康铜	0.789	1.611	2.467	3.357	4.277
镍铬 - 康铜	1.192	2.419	3.683	4.983	6.317
铁 - 康铜	1.019	2.058	3.115	4.186	5.268
镍铬 - 铝镍	0.798	1.611	2.436	3.266	4.095
铂 - 10% Rh，Pt	0.113	0.235	0.365	0.502	0.645

6.6　设计一个测量系统，可用于在平面上分散的多个点处监测温度，温度最高可达 100℃，然后把结果显示在控制面板上。

6.7　设计测量容器中的液位，容器中有一个浮漂，在竖直方向上可使一个悬臂梁弯曲。悬臂的弯曲程度用做衡量液位的高低。当力 F 施加到长度为 L 的悬臂梁自由端时，表面应变和固定端的距离 x 之间的关系为：

$$\text{应变} = \frac{6(L-x)}{wt^2E}$$

其中，w 是悬臂的宽度，它的厚度是 t，材料的弹性模量是 E。应变片用于监测悬臂梁的弯曲，将 2 个应变片分别安装在上表面的长度方向上，2 个分别安装在下表面的长度方向上。该应变片使用了四臂的惠斯通电桥，用输出电压（可能被放大）测量液位。如果液位每变化 10cm 输出 10mV 电压，请确定这个系统部件需要的指标。

161

6.8　设计一个基于 40mm 直径膜片传感器的静态压力测量系统，可承受最大 500MPa 的压差。该膜片的中心形变 y 远小于厚度：

$$y \approx \frac{3r^2P(1-v^2)}{16Et^3}$$

其中，r 是膜片的半径，P 是压差，E 是弹性模量，v 是泊松比。解释形变 y 如何被转换成一个信号，并且可以显示在一个仪表上。

6.9　给出下列测量系统可能考虑使用的元件：

(a) 检测压缩空气管路的气压，把结果显示在表盘上，没有很高的精度要求。

(b) 连续监测并记录室内温度，精度为 ±1℃。

(c) 监测通过称重平台的卡车的重量。

(d) 监测旋转轴的角速度。

162

第三部分 驱 动

第7章 气动与液压驱动系统

本章目的

通过本章的学习，读者将能够：

- 看懂系统图纸，设计包含液压/气动方向控制阀和缸体的简单顺序控制系统。
- 解释过程控制阀的原理、特点和选型计算。

7.1 驱动系统

驱动系统是控制系统的一个组成部分，负责将一个微处理器或控制系统的输出变换成一台机器或设备的控制动作。举例来说，我们可以从控制器中得到电输出，这个电信号输出必须转换成直线运动，以移动负载。再比如说，从控制器得到电输出，也可以转变为控制一个管道的液体流量的动作。

在这一章中我们对流体动力系统，即气动和液压驱动系统进行了讨论。使用压缩空气的术语，称为**气动技术**；使用液体的术语（典型的如液压油），称为**液压技术**。第8章讨论机械驱动系统，第9章讨论电动驱动系统。

7.2 气动与液压系统

气动信号经常用来控制终端控制元件，即使控制系统本身是电气控制系统。这是因为气动信号可用于驱动大型阀门和其他高功率的控制设备去移动较大的负荷。然而气动系统有一个主要缺点，即空气的可压缩性。液压系统可以用于更高功率的控制装置，但比气动系统更昂贵，而且油泄漏会带来一定危害，空气泄漏却不会。

大气压力变化取决于位置和时间，但在气体力学中，一个大气压俗称1bar（巴），一般约等于10^5Pa。

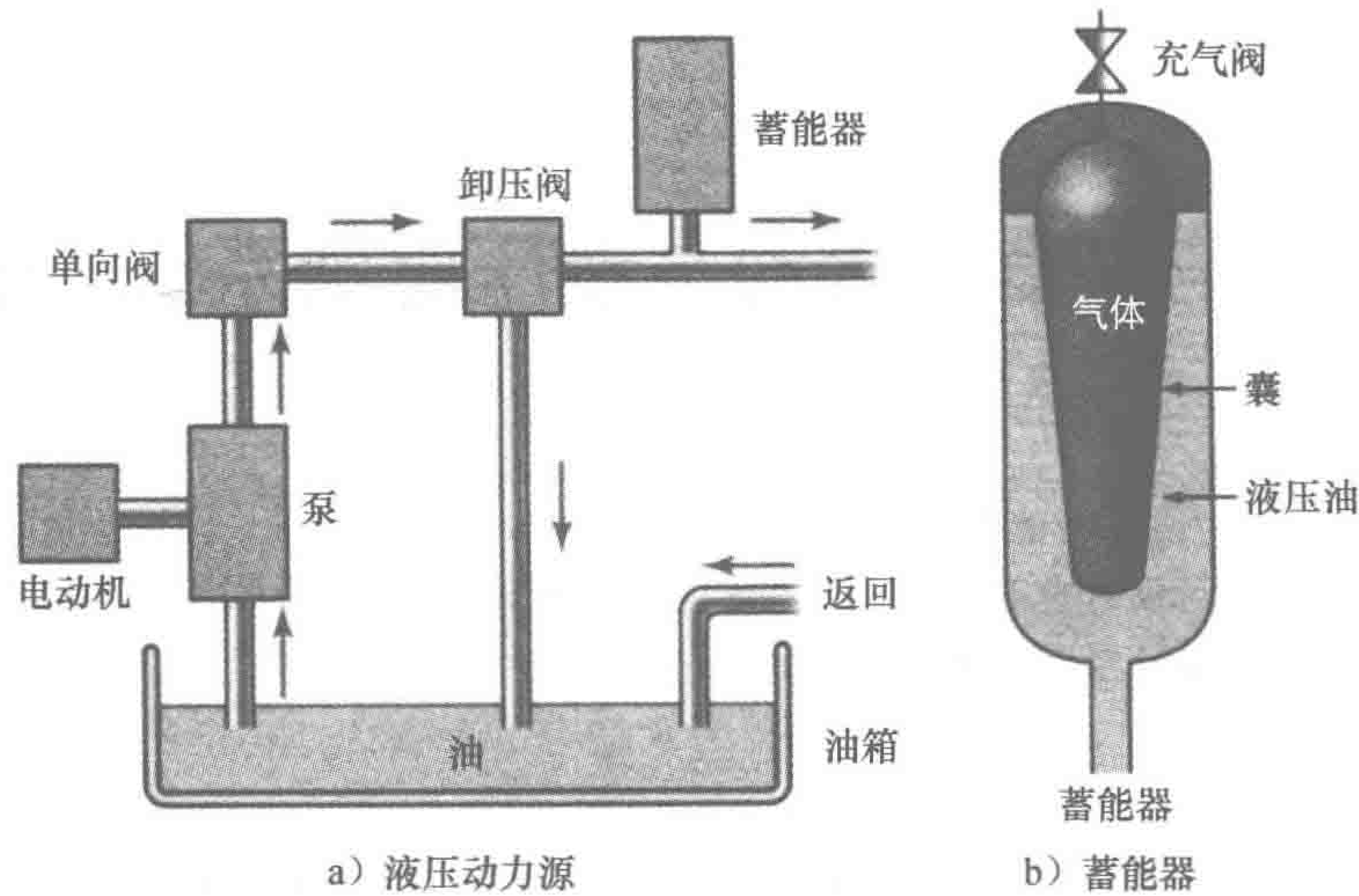
7.2.1 液压系统

在液压系统中，压力油由一个电动机驱动的泵提供。泵油从油箱通过一个单向阀和一个蓄能器输送到系统，再沿着它返回到油箱。图7-1a展示了一种布置方式。如果压力高于一个安全的水平，那么卸压阀释放压力。单向阀可以防止油被反向驱动，而蓄能器可以消除任何瞬时的输出油压的波动。从本质上讲，蓄能器只是一个使油压保持在一定压力下的容器，图7-1b表示的是蓄能器的常用形式，在充满液压液体的腔内有一个充满压力气体的囊，旧的类型包含一个弹簧加载的活塞。如果油的压力上升，囊状体收缩，增加油占据的体积，从而减少压力。如果油的压力下降，囊状体扩大，以减少油占据的体积，因此增加压力。

163 ~ 165

常用的液压泵包括齿轮泵、叶片泵和活塞泵。**齿轮泵**包括两个旋转方向相反的紧密啮合的齿轮（见图7-2a）。流体被挤压进旋转的齿轮齿和壳体之间，接着从进口挤到出口，再被排

出。因为成本低、稳定性高，所以这种泵被广泛使用。它们一般工作于低于约 15MPa（150 巴）的压力下，转速低于 2400r/min，最大流量约为 0.5m^3/min。但是，在齿和壳体之间以及啮合齿之间会发生泄漏，而这限制了效率。**叶片泵**的转子槽上装有由弹簧加载的滑动叶片（见图 7-2b），当转子旋转时，叶片沿着壳体的轮廓转动。这使得流体被挤入相邻的叶片和壳体之间，并借此把液体从入口传送到出口。叶片泵的泄漏小于与齿轮泵。**柱塞泵**在液压中的应用可以采取许多



a）液压动力源　　b）蓄能器

图 7-1　液压系统

a）齿轮泵　　c）径向柱塞泵

b）叶片泵　　d）斜盘式轴向柱塞泵

图 7-2　液压泵

形式。对于**径向柱塞泵**（见图 7-2c），缸块围绕着固定凸轮旋转，造成活塞缸的空隙，通过弹簧复位，使活塞向内或者向外运动。其结果是，流体在进入口被吸入，旋转一定角度后从排出口喷射出。对于**轴向柱塞泵**（见图 7-2d），其活塞为轴向移动而非径向。活塞被轴向布置在一个旋转的缸体内，并使之与旋转斜盘接触联动。滑靴板与驱动轴有一定角度，因此，当轴旋转时，滑靴板带动活塞移动。当活塞与进入口反向时，空气被吸入，而当与排出口反向时，空气被排出。活塞泵具有较高的效率，并且适用于比齿轮泵或叶片泵更高的液体压力场合。

7.2.2　气动系统

带有**气动**功率源（见图 7-3）的电动机可以驱动空气压缩机。空气压缩机的进气口可以进行过滤，并且通过消音器降低噪声。卸压阀可提供保护，防止在系统中的压力大于安全压力。因为空气压缩机会提高空气的温度，所以可能需要冷却系统。另外带有脱水器的过滤器也是需要的，以便去除空气中的污染物和水分。空气接收器可以增加系统中空气的体积，并可以平滑任意短期的压力波动。

166～167

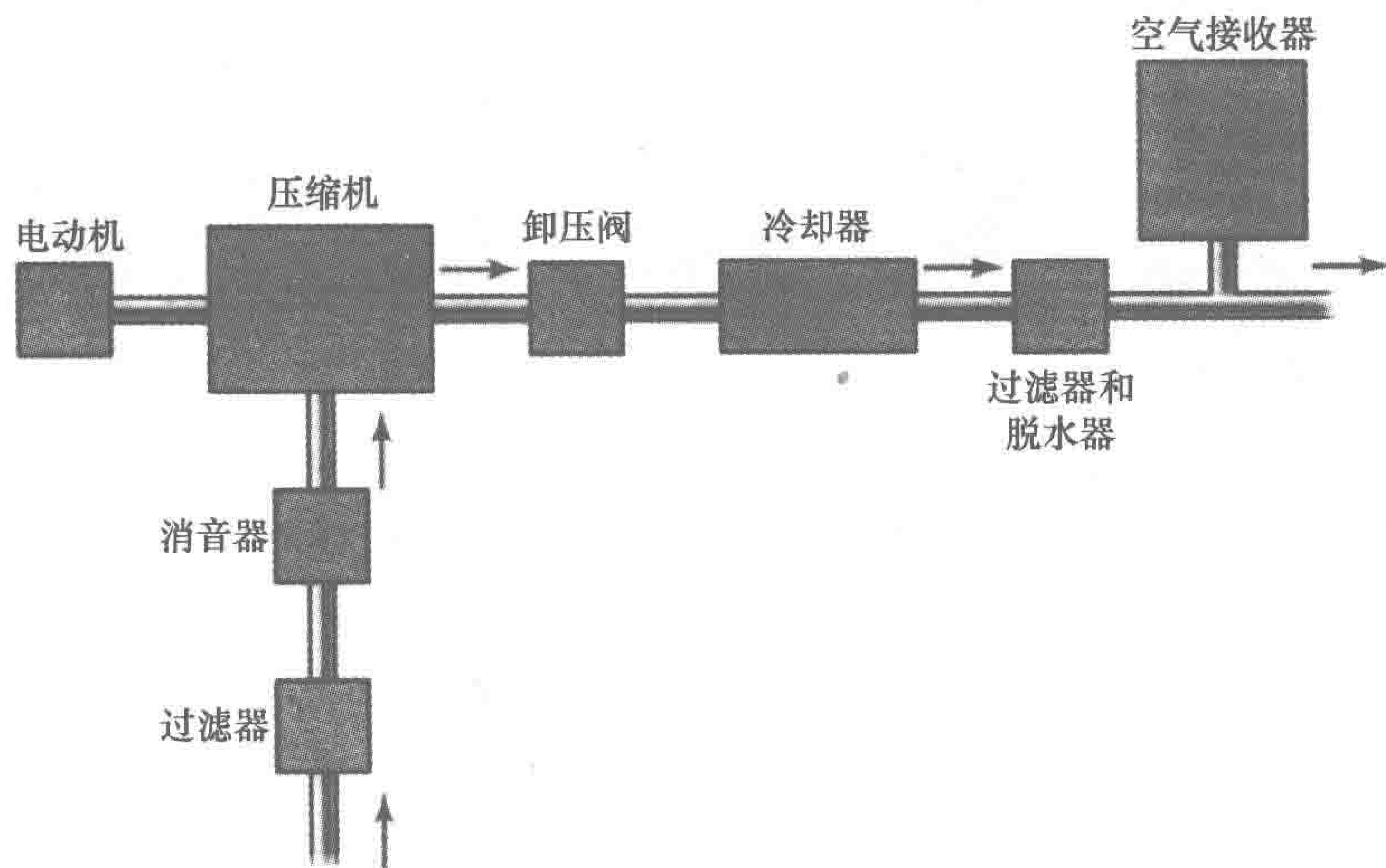
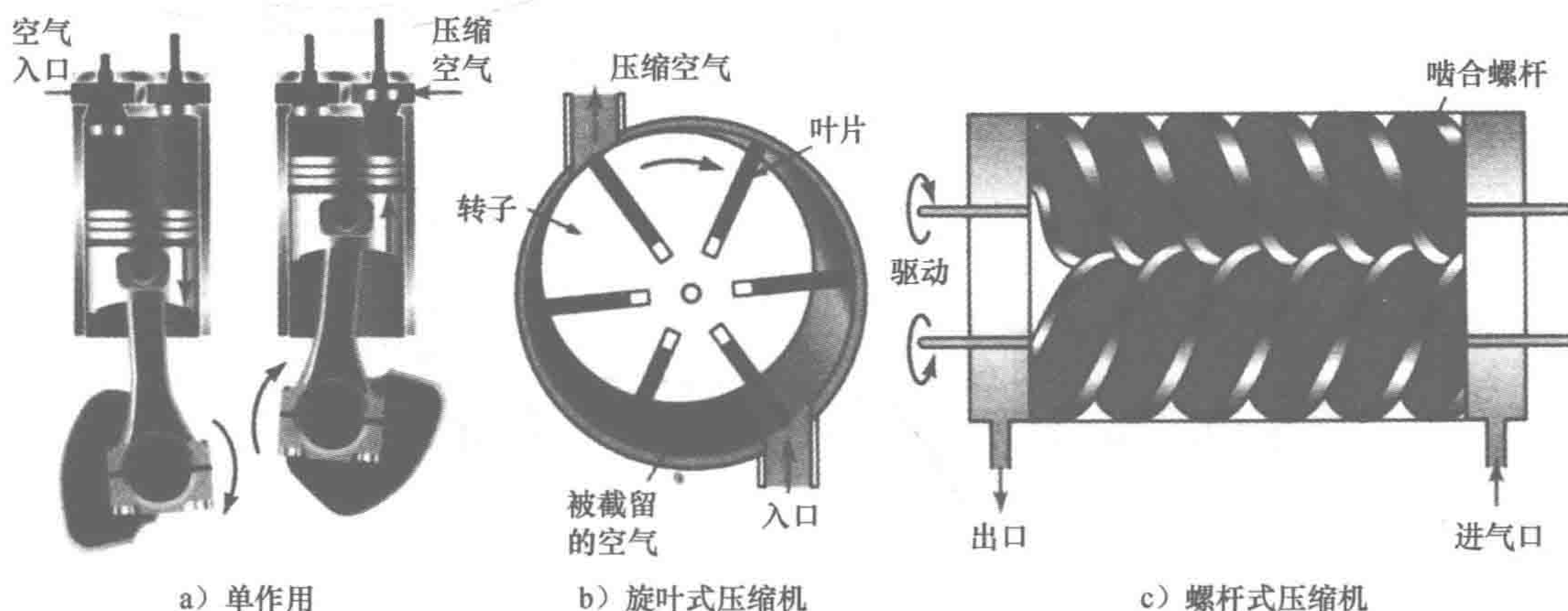


图 7-3　气动电源

常用的空气压缩机是在将连续体积的空气分离后压缩。图 7-4a 给出了一个单动式、单级、垂直、往复式压缩机的基本形式。在进气冲程中，降低活塞，通过弹簧加载的进气阀使空气吸入该腔室。活塞开始再次上升，截留的空气强制关闭进气阀来完成压缩。当空气压力上升足够时，弹簧加载的排出阀打开，截留的空气流入压缩空气系统。在活塞到达上死点中心时，即开始下降，以上过程不断重复循环。这种压缩机称为**单作用的**，因为每个活塞冲程产生压缩气体，所以**双作用**压缩机被设计成在活塞向上和向下过程中都产生气体。在单次操作过程中，压缩机直接将气体压力从大气压力提升到期望的压力，所以称之为**单级**。当产生超过几巴[⊖]的压缩空气时，两级或者多级更常用。通常情况下，两级压缩适用于产生大约 10 ~ 15 bar 的压力，更高的压力就需要使用多级压缩。因此，一个具有两级的压缩机，我们可以在第一级把大气压力下的空气压缩到 2 bar，然后第二级压缩该空气到7 bar。往复式压缩机也可以用做一个单级压缩机以产生高达约 12 bar 的空气压力；或者作为一个多级压缩机，产生高达约 140 bar 的压力。通常情况下，气流的流量变化范围从约 0.02m^3/min 自由空气到约 600m^3/min 自由空气。自由空气是指在正常大气压下的空气。压缩机的另一种

⊖　1bar = 10^5Pa。——编辑注

形式是**旋叶式压缩机**。压缩机转子偏心地安装在圆筒形腔室内（见图7-4b）。该转子具有桨片和叶片，它可在径向槽内自由地滑动，旋转引起叶片贴着气缸壁运动。当转子旋转时，空气被困在由叶片形成的气囊内，转子旋转使气囊变小，空气被压缩。因此，压缩空气从排出口排出。单级旋叶式压缩机通常可以用于压力高达约800kPa，流速从0.3～30m³/min自由空气的输送。压缩机的另一种形式是**旋转螺杆式压缩机**（见图7-4c）。它由两个在相反方向旋转啮合旋转的螺杆组成。由于螺杆的旋转，通过进气口和壳体之间的螺杆，空气被吸入壳体，进入螺杆之间的空隙中。被截留的空气沿着螺杆的长度压缩，该空间逐渐变小，从而从排出口排出。通常，单级、旋转式的螺杆压缩机可用于压力大约1000kPa、流速1.4～60m³/min自由空气的输送。



a）单作用　　b）旋叶式压缩机　　c）螺杆式压缩机

图7-4　常用的空气压缩机

7.2.3　阀

阀用于液压和气动系统中流体流量的引导和调节。阀一般有两种形式，**有限位置阀**和**无限位置阀**。有限位置阀仅仅执行允许或阻止流体流动的动作，因此可用于打开或关闭开关驱动器，可用于方向控制，使流体从一个路径切换到另一个，从一个驱动器切换到另一个。无限位置阀能够在全开和全关之间的任意位置控制流体，因此用于控制改变驱动器力或过程控制情况下的流体流速。

7.3　方向控制阀

气动和液压系统使用方向控制阀来控制整个系统的流体流动。它们不能用于改变流体的流动的速率，但可以使之完全打开或完全关闭，即开/关设备。这样的开/关阀广泛用于开发顺序控制系统（见7.5.1节）。它们可通过机械、电气或流体压力信号被激活，继而切换流体的流动方向。

168～169

常见的方向控制阀类型是**滑阀**（spool valve），其阀芯在阀体内水平移动来控制流量。图7-5给出了一种特殊形式。在图7-5a中，空气供给被连接到端口1时，端口3被关闭。因此，连接到端口2的设备可以被加压。当阀芯移动到左侧（见图7-5b）时，空气供给被切断，端口2被连接到端口3。端口3通到大气中，所以连接到端口2的系统中的空气压力也通到大气中。因此，阀芯运动先使空气流进系统，然后反向使空气流出该系统。**转阀**有一个转动的阀芯，当它旋转时，会以类似的方式打开和关闭端口。

另一种常见的方向控制阀是**提升阀**（poppet Valve）。图7-6显示的是其中一种形式。该阀通常是处在封闭的状态下，即与压力供给相连的端口1和与系统相连的端口2之间是隔断的。

在提升阀中，球、圆盘或椎体，与阀座一道控制流量。在图 7-6 中所示为球体。当按钮按下时，球被压出其座位，端口 1 连接到端口 2，使流体流动。当按钮被释放时，该弹簧迫使球回到它的位置，流动停止。

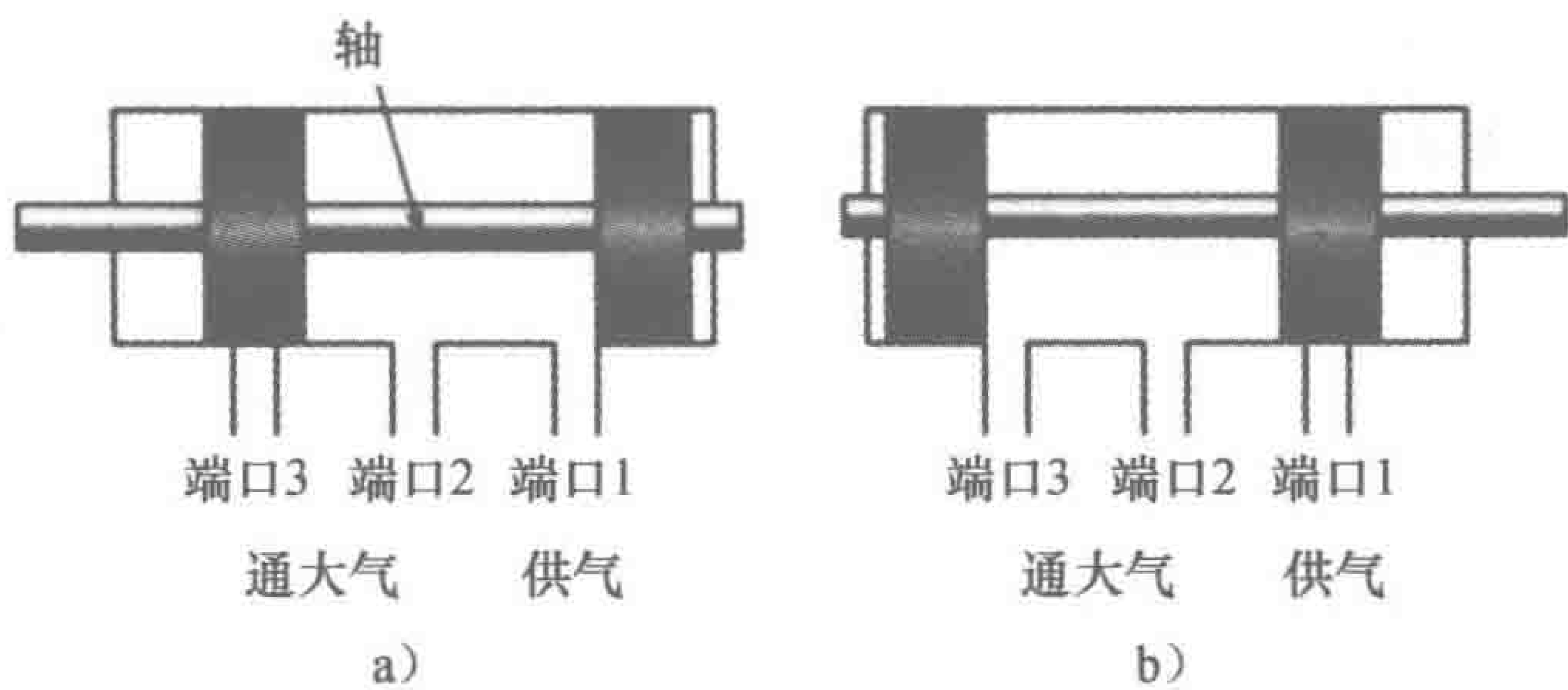


图 7-5　滑阀

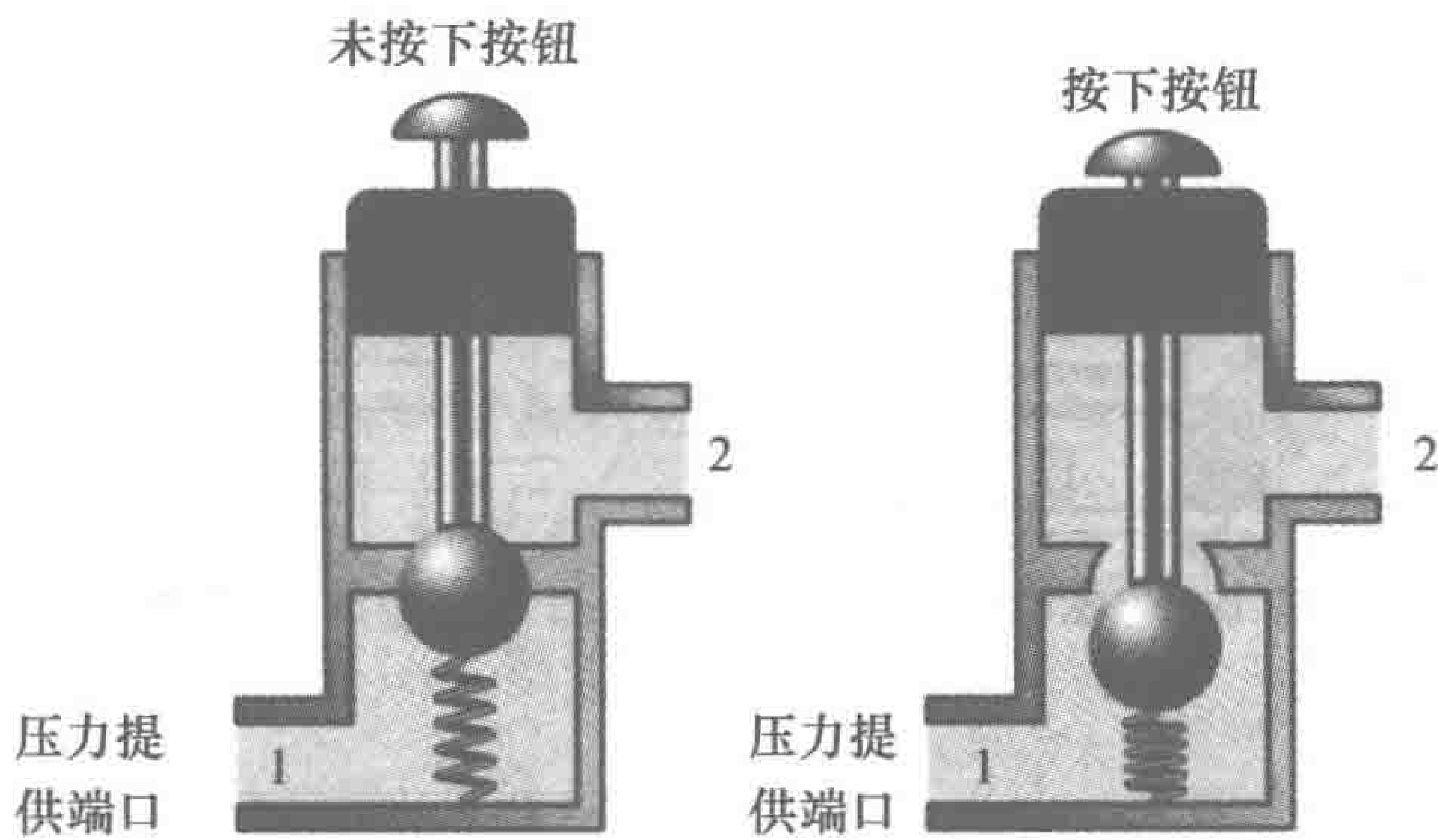
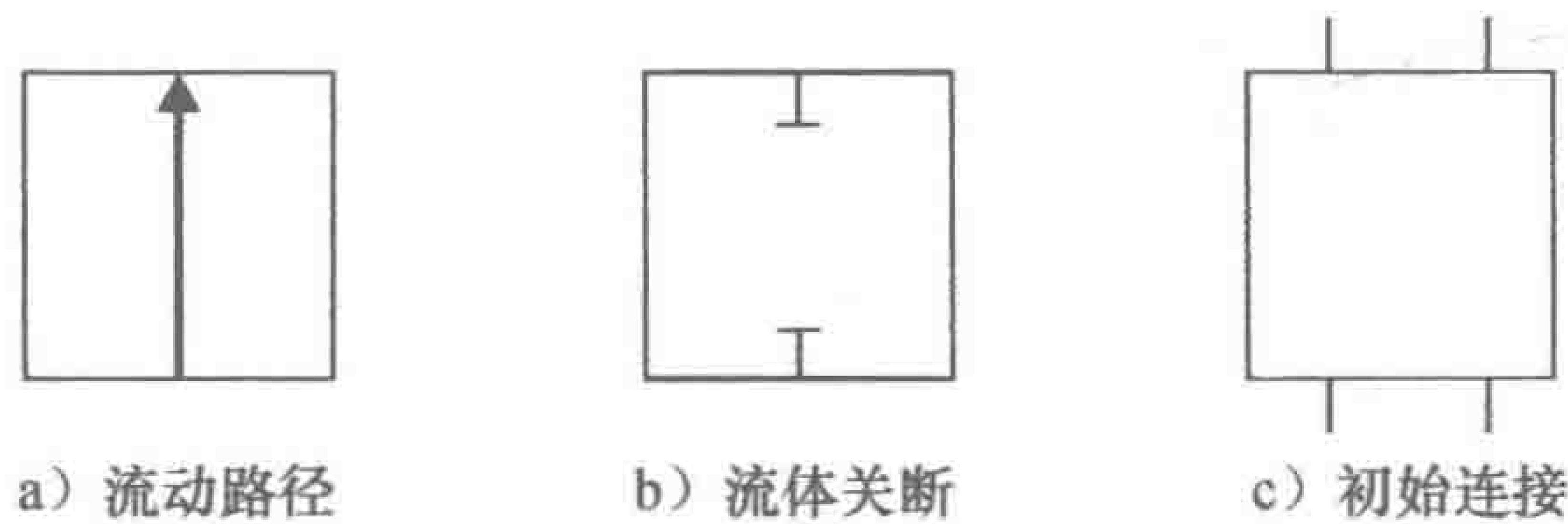


图 7-6　提升阀

7.3.1　阀的符号

控制阀的符号由一个正方形来表示其每个开关位置。因此，对于图 7-6 所示的提升阀，有两个位置：一个是按钮没有被按下，另一个是按钮被按下。因此，一个二位阀将有两个正方形，一个三位阀有三个正方形。带箭头的线（见图 7-7a）用来指示每个位置中的流动方向，中断截止线表示关闭的流体（见图 7-7b）。阀初始位置连接的端口如图 7-7c 所示，图中该阀具有四个端口。端口根据它们的功能标记为数字或字母。标记为 1（或 P）的端口为压力供给端口，3 或 5（或 R 或 S）为排气端口，3（或 T）为液压返回端口，2 或 4（或 B 或 A）为输出端口。 170



a）流动路径　b）流体关断　c）初始连接

图 7-7　符号

图 7-8a 给出了用各种符号代表驱动阀的各种方式。当然，不仅仅是这些符号可以作为阀符号。举一个例子，图 7-8b 给出的符号是图 7-6 中的两端口的二位提升阀。需要注意的是一个两端口的二位提升阀将被描述为一个 2/2 阀，第一个数字表示端口数量，第二个数字表示位置数量。阀驱动是由一个按钮和一个弹簧完成的。

为了进一步的说明，图 7-9 给出了一个螺线管控制的滑阀和它的符号。该阀由通过螺线管

的电流驱动，并通过弹簧返回到其原始的位置。

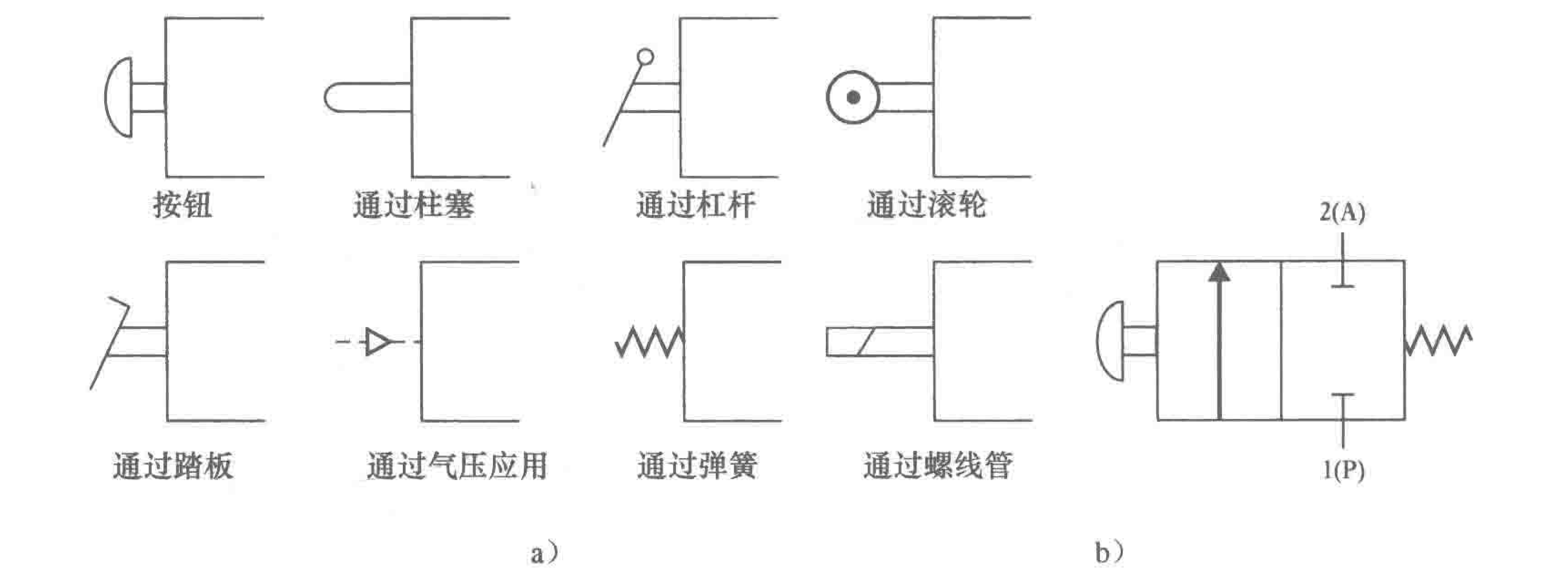


图 7-8　阀门驱动符号

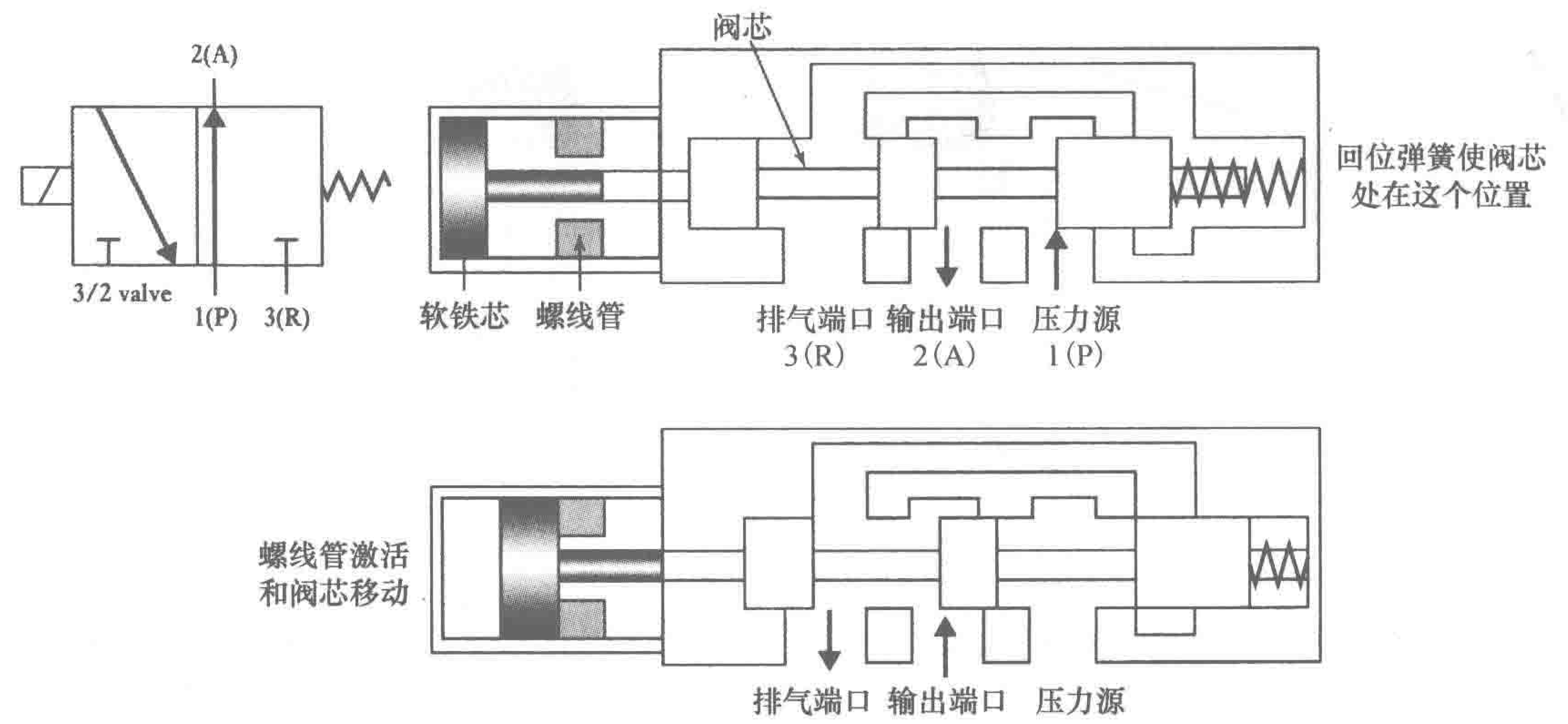


图 7-9　单螺线管阀

图 7-10 显示了一个 4/2 阀的符号。连接图中表示出了初始状态，即 1(P) 被连接到 4(A)，3(R) 关闭。当螺丝管被激活时，与初始状态相连的正方形里的符号所示的状态即为阀的状态，即 1(P) 连接到 2(B)、4(A) 连接到 3(R)。当通过螺线管的电流停止时，弹簧推压阀芯返回到其初始位置。弹簧运动的状态用正方形里连接的符号来表示。

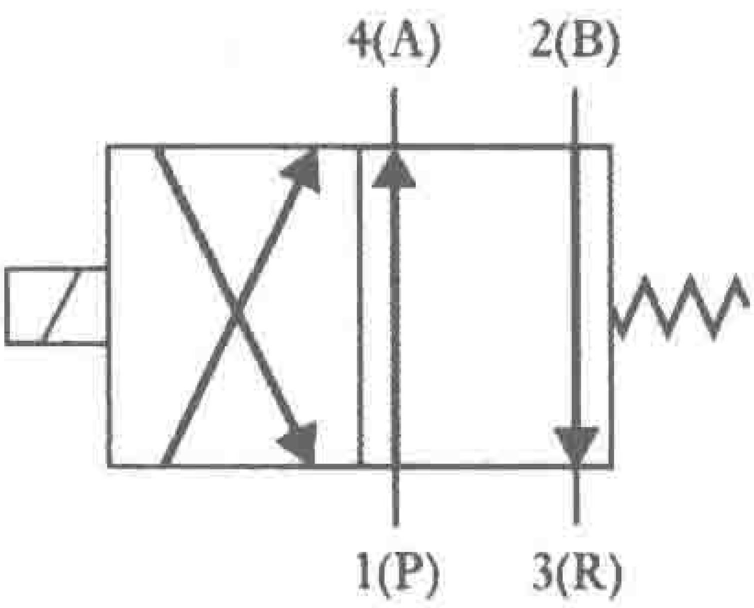


图 7-10　一个 4/2 阀

图 7-11 为阀应用于气动升降系统中的一个简单例子。使用两个按钮的 2/2 阀。当向上阀的按钮被按下时，负载被提高。当向下阀的按钮被按下时，负载被降低。注意，气动系统使用一个空心箭头来表示排气到大气中。

7.3.2　先导阀

移动阀体中的球或传送装置所需的力对手动或螺线管操作来说往往太大了。为了克服这个问题，就使用**先导阀**，其中第一个阀用来控制第二个阀。图 7-12 说明了这一点。先导阀容量小，并且可由手动或螺线管操作。它可使主阀受系统压力的控制。先导压力线用虚线来表示。先导阀和主阀可以是两个独立的阀，但更多情况下，它们结合在单个壳体中。

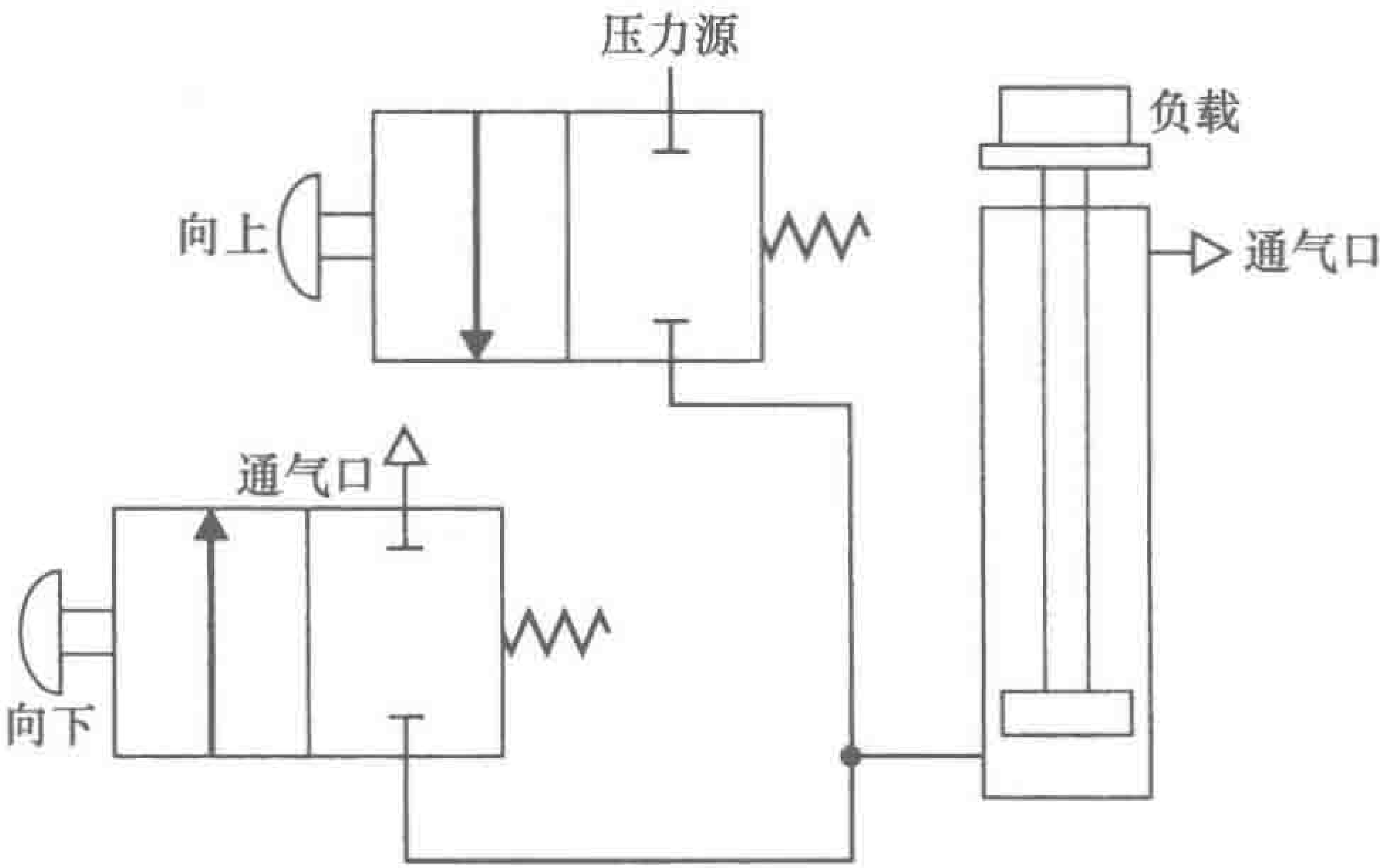


图 7-11　提升系统

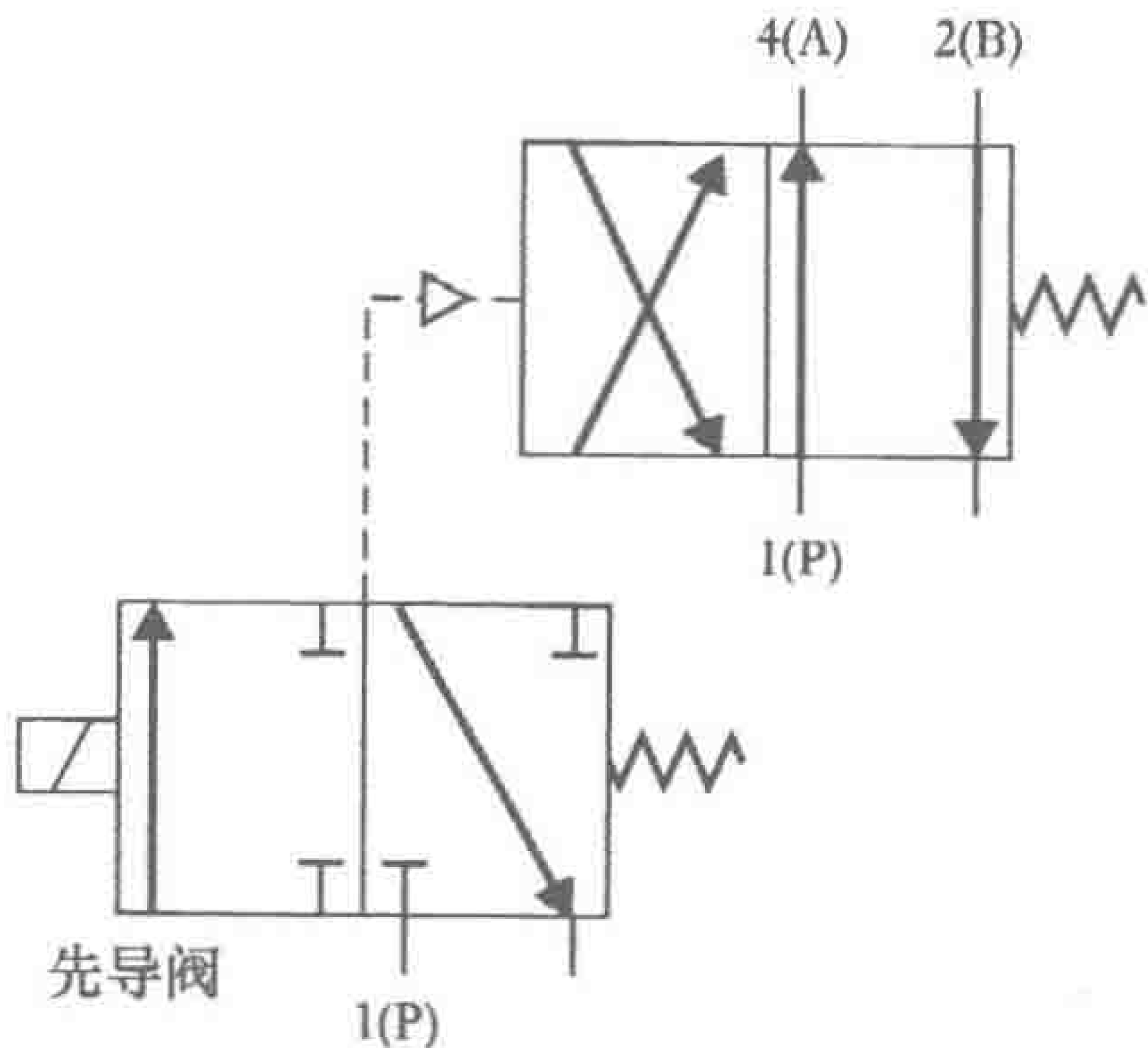


图 7-12　先导式系统

7.3.3　单向阀

图 7-13 显示了一个简单的单向阀以及它的符号。在球被按压在弹簧上时，流体单方向流过阀。在另一个方向上，因为弹簧迫使球贴住阀座，所以流动被阻止。

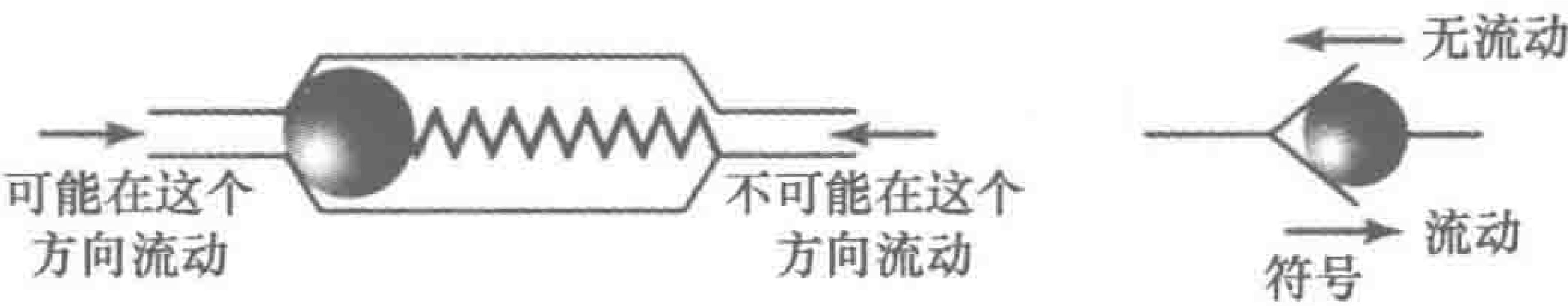


图 7-13　单向阀

7.4　压力控制阀

压力控制阀主要有三种类型：

1. 压力调节阀

被用来控制环路中的操作压力，并维持在一个恒定值。

2. 限压阀

用做安全装置，以限制系统中的压力低于某个安全值。如果压力上升超过设定的安全值，阀门打开，泄放到大气中，或返回到储槽。图 7-14 示出了**限压阀/泄压阀**，它有一个孔，通常是关闭的。当入口压力克服弹簧施加的力时，阀打开，气压由通气口泄放到大气中，或返回到储槽。

171~173

3. 压力顺序阀

这些阀门用于感测外部线路的压力，并在它达到一定预设值时给出一个信号。图 7-15 中的限压阀，将压力设为阀的入口压力。我们可以将其改造成顺序阀，以便允许压力上升到所要求的水平时，系统的某些部分可以发生流体流动。例如，在自动化机器中，启动时可能需要向工件上施加一定的夹紧力。图 7-15a 显示了顺序阀的符号，当入口压力达到一定值时，阀门打开并将压力施加到后续的系统。图 7-15b 显示了使用这样的顺序阀的系统。首先，当 4/3 阀操作时，压力被施加到气缸 1，它的柱塞移动到右边。当压力过小时，无法操作顺序阀，所以没有压力施加到缸体 2。当气缸 1 的柱塞到达端部挡块时，系统中的压力上升，达到一个适当的水平后，触发顺序阀打开，继而施加压力

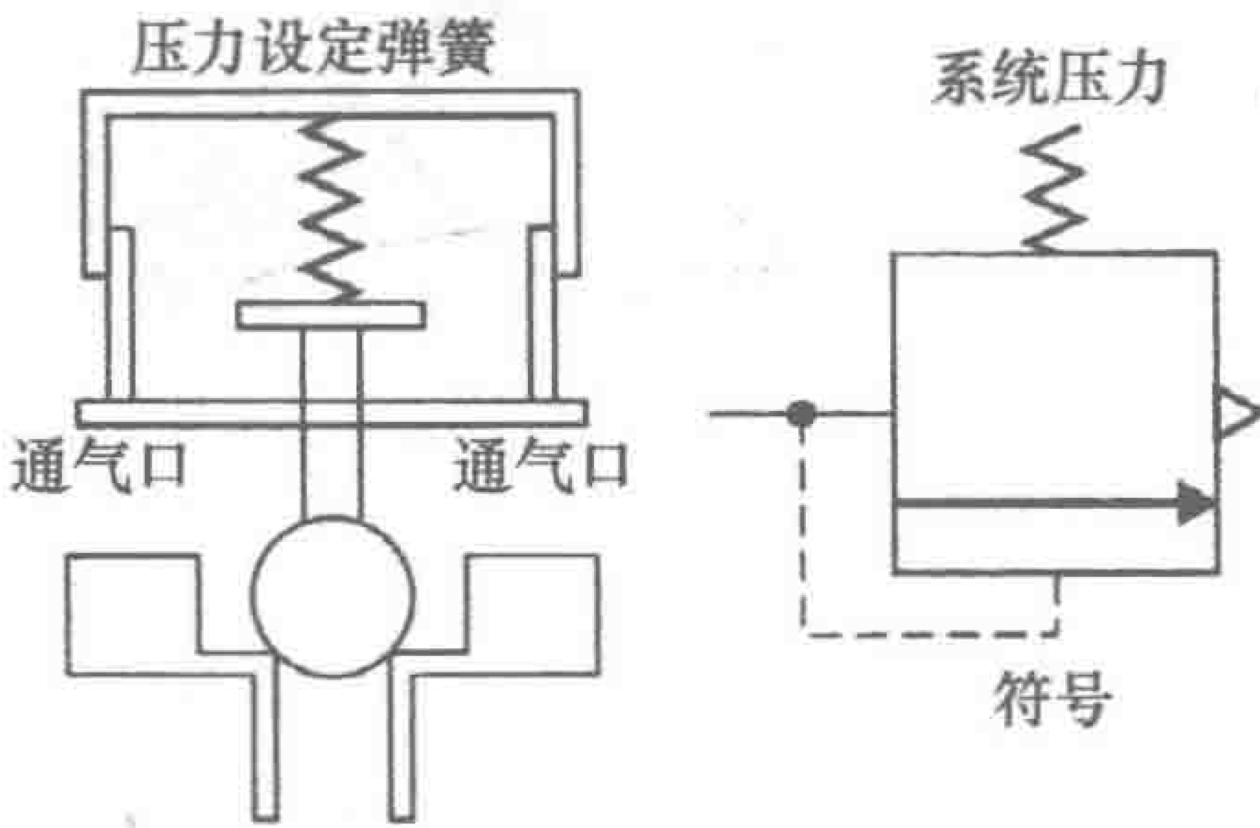
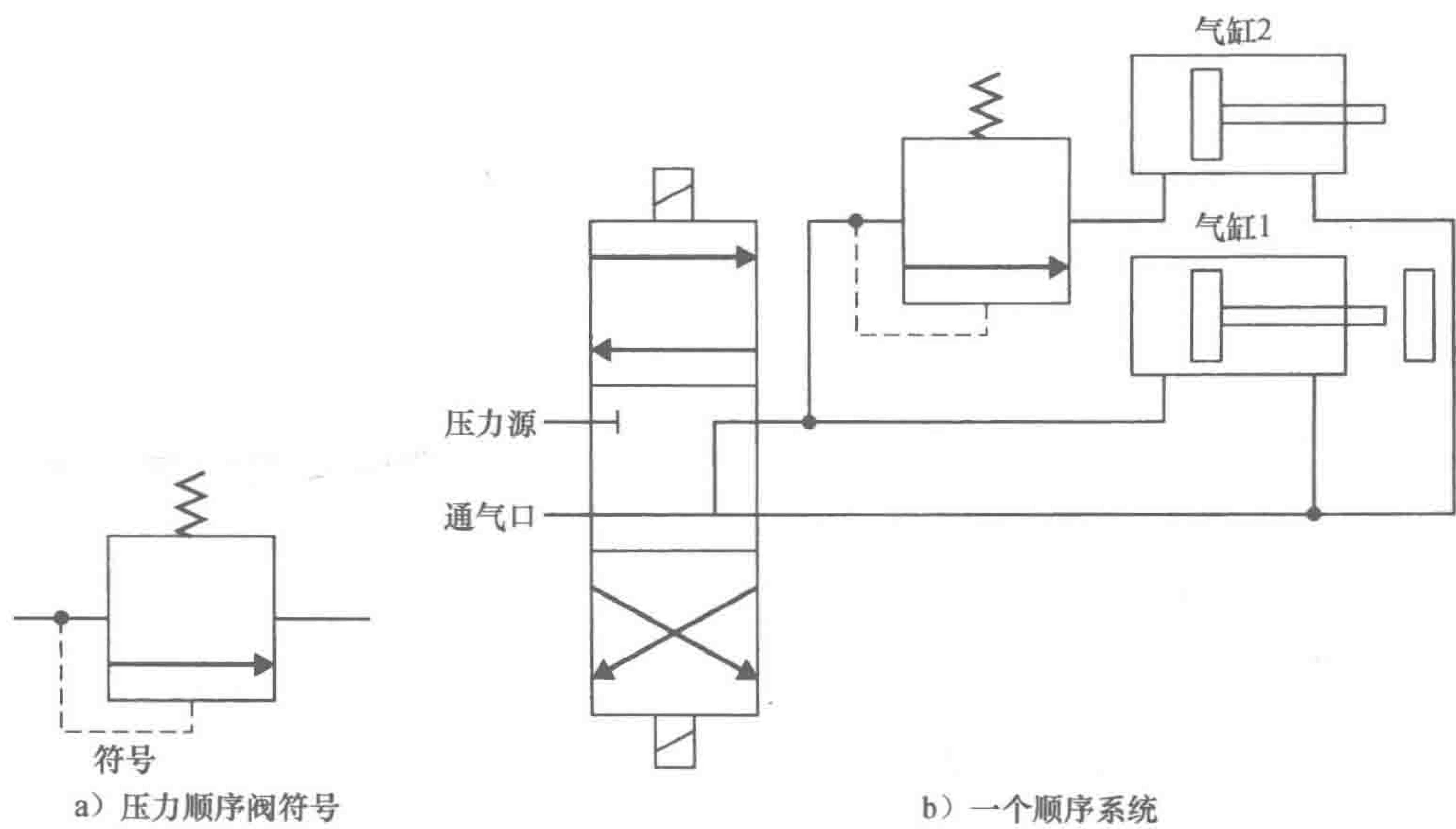


图 7-14　限压阀

到气缸 2，使其柱塞开始运动。



174

图 7-15　顺序阀符号与例子

7.5 缸体

液压缸或**气缸**是一个线性驱动器的例子。对于液压和气动，其原理和形式是相同的，不同之处是液压系统的压力较高而引起尺寸大小的问题。缸体包括一个圆筒形管，活塞/柱塞可沿该管来回滑动。缸体有单作用缸和双作用缸两种基本类型。

当被控制的压力施加到活塞的一侧时，单作用缸将会使用，弹簧往往用来阻碍活塞的运动。活塞的另一侧通向大气。图 7-16 显示了这样一个具有弹簧复位的气缸。以流体压力 p 施加在活塞的一侧，另一侧是在大气压力下，这样便在活塞上产生了一个大小为 pA 的力，其中 A 是活塞的面积。在活塞杆上的实际作用力将由于摩擦而变小。

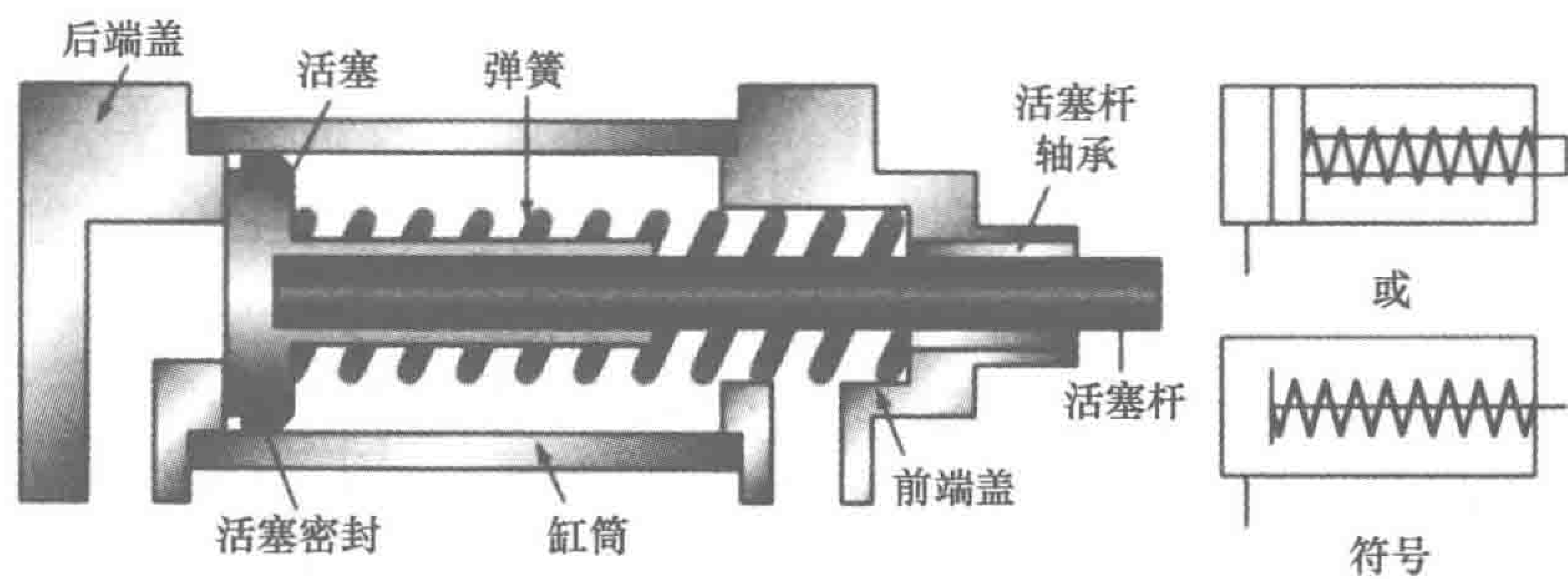


图 7-16　单作用气缸

对于图 7-17 中所示的单作用气缸，当电流通过螺线管时，阀切换位置，并将压力施加到活塞使之沿着缸移动。当通过螺线管的电流消失时，该阀将恢复到其初始位置，空气从缸体排出。其结果是弹簧使活塞沿着缸体返回。

当控制的压力施加到活塞的两侧时，使用“双作用”术语（见图 7-18）。活塞两侧压力的不同导致活塞的移动。活塞将沿着缸体朝由高压信号决定的方向移动。对于图 7-19 中所示的双作用缸，通过一个螺线管的电流，使活塞向一个方向移动，通过另一个螺线管的电流使活塞

沿相反方向移动。

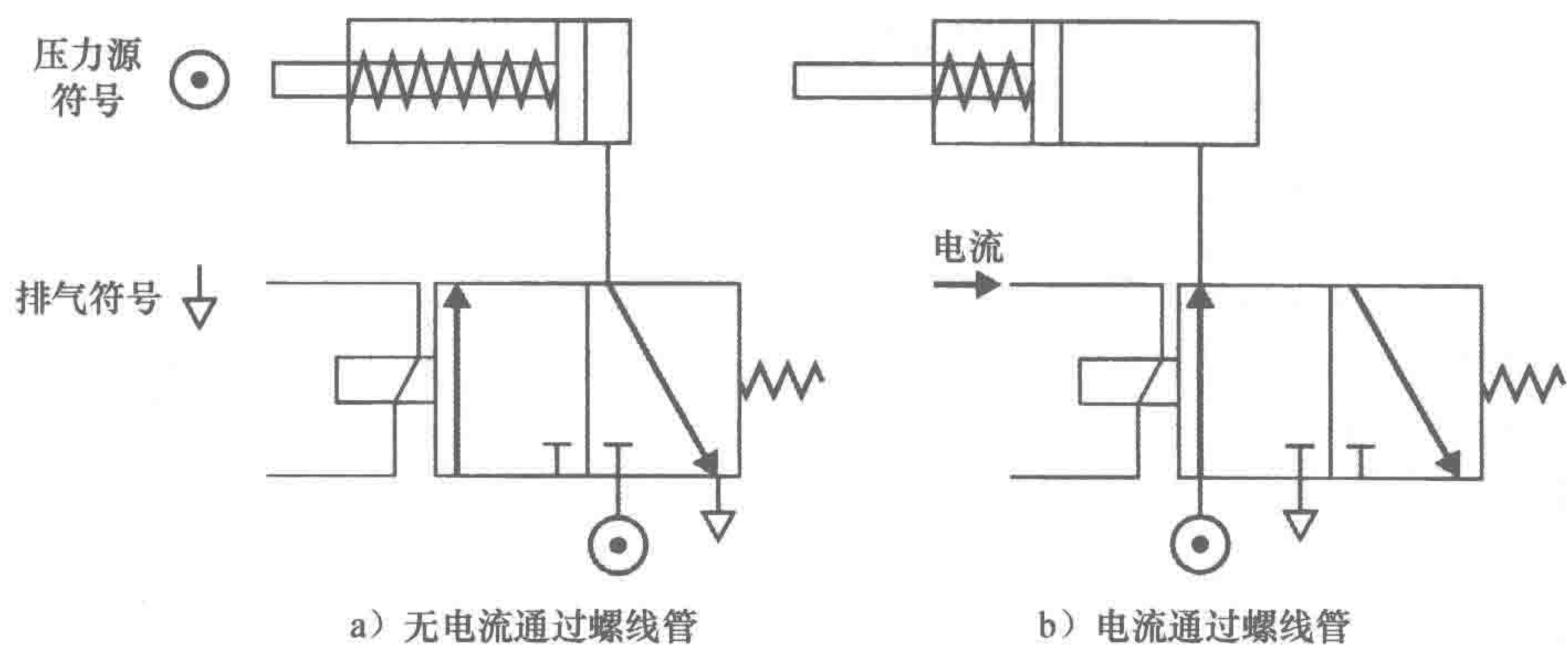


图 7-17　单作用气缸的控制

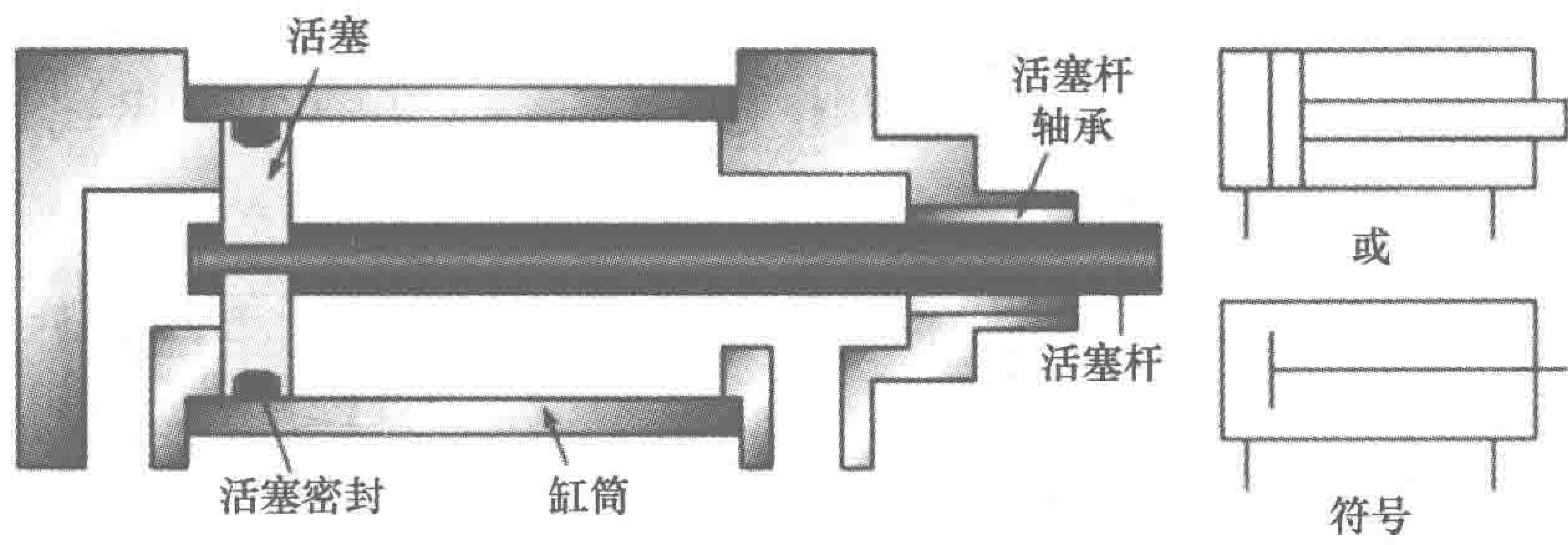


图 7-18　双作用气缸

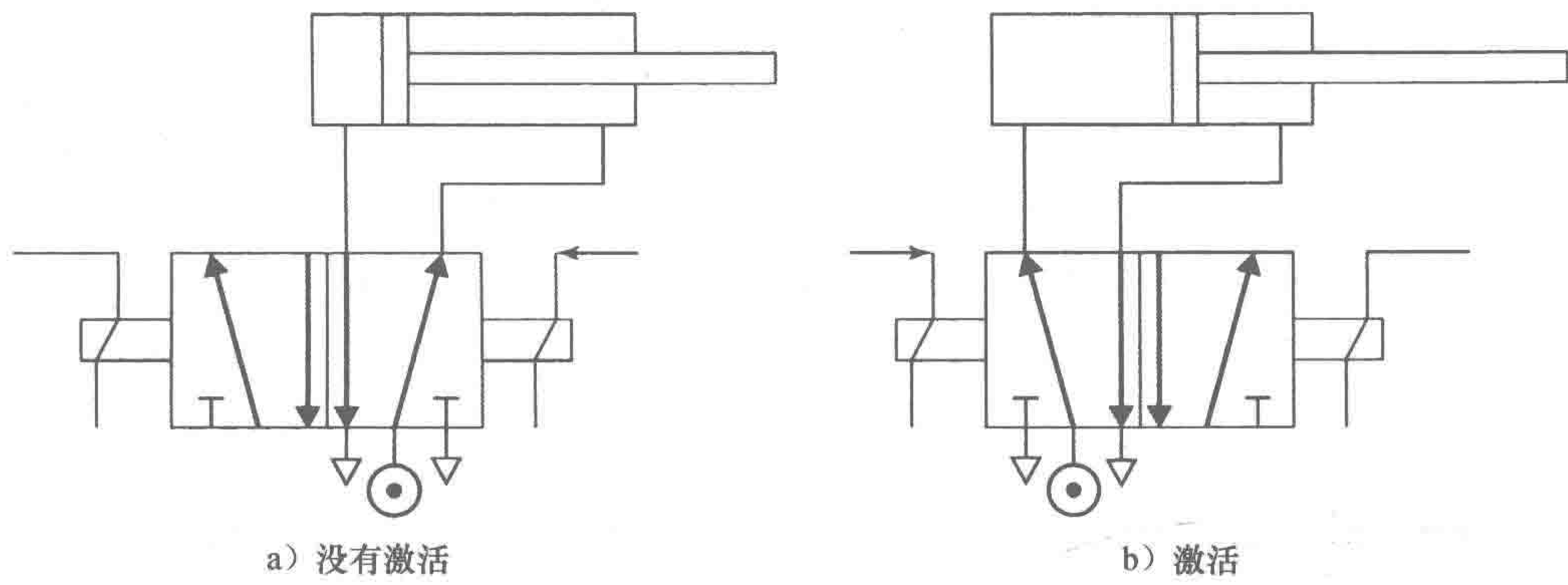


图 7-19　使用螺线管控制双作用气缸

缸体的选择取决于移动负载所需的力和所需的速度。液压缸比气缸可以提供更大的动力。但气缸能够达到更快的速度。缸体产生的力等于缸体的横截面积与工作压强的乘积，换言之，即缸体内活塞两侧压强之差。一个工作压强为 500kPa、直径为 50mm 的气缸，将会产生 982N 的力。具有相同直径和 15 000kPa 工作压强的液压缸，会产生 29.5kN 的力。

如果液压油流入液压缸的流量是每秒Q，那么被活塞挤出的液压油的流量也是每秒Q。对于一个横截面积为 A 的活塞，其速度为 v，则可得$Q = Av$。因此，液压缸的速度 v 就等于通过液压缸的流量除以液压缸的横截面积 A。所以对于一个直径为 50mm 的液压缸，其液压油流量为 $7.5\times10^{-3}\mathrm{m}^3/\mathrm{s}$，速度则为 3.8m/s。气缸的速度则不能以这样的方式来计算，因为其速度依赖

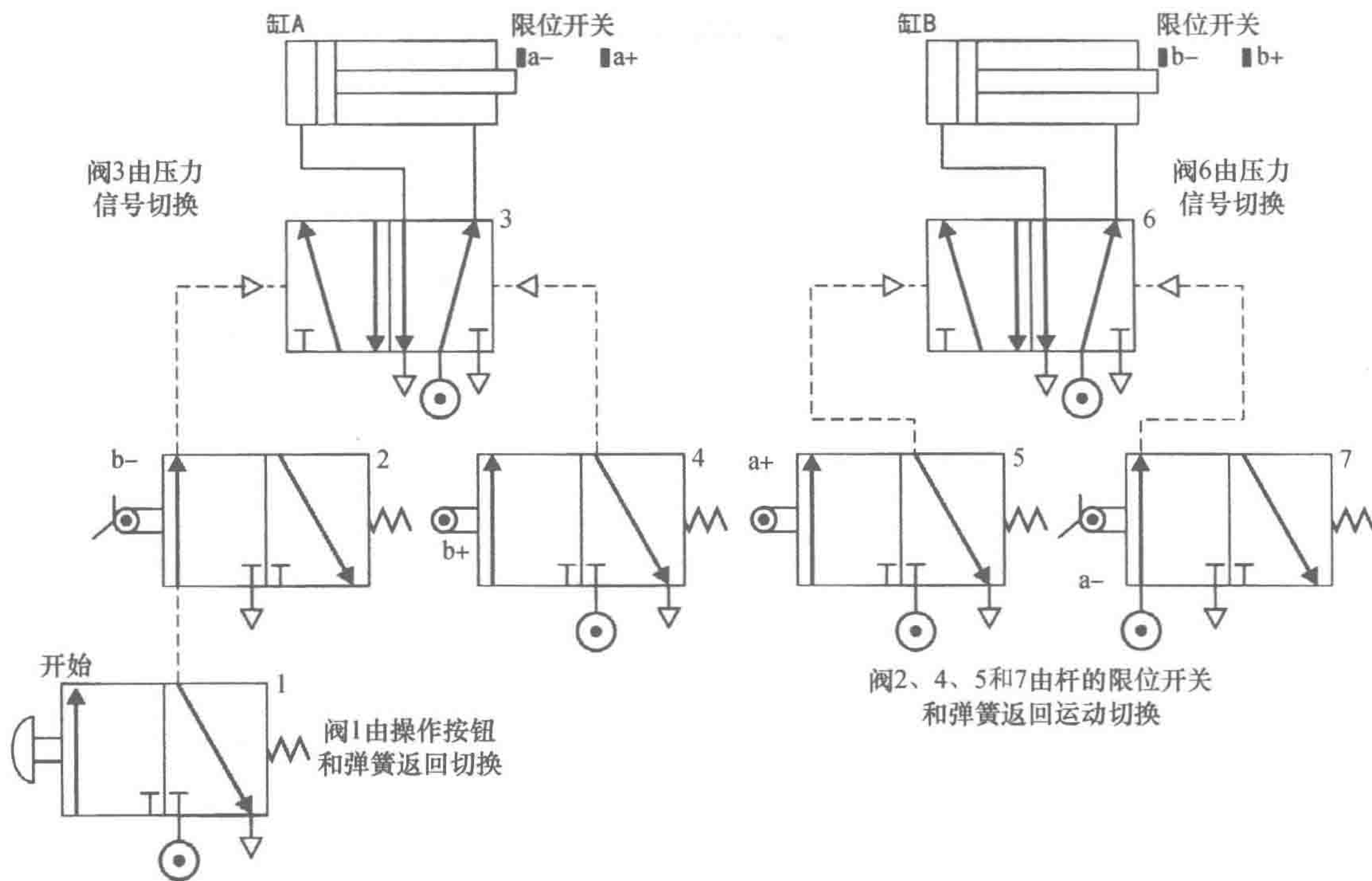
175~176

于活塞前面空气排出的速率。速度的调节可以通过阀来实现。

为了阐明上述内容，考虑一个在加工环境中使用液压缸在15s内使工件移动250mm的问题。如果推动这个工件需要50kN的力，那么在带有活塞直径为150mm的活塞的液压缸中，需要的工作压强和液压油的流量为多少时才能满足条件？横截面积为$\frac{1}{4}\pi \times 0.150^2 = 0.0177\mathrm{m}^2$。由液压缸产生的力为液压缸的横截面积和工作压强的作用结果。所以，工作压强为$50 \times 10^3 / 0.0177 = 2.8\mathrm{MPa}$。液压缸的速度等于通过缸的流体的流量与缸的横截面积之比。因此需要的流量为$(0.250/15) \times 0.0177 = 2.95 \times 10^{-4}\mathrm{m}^3/\mathrm{s}$。

7.5.1 缸的顺序控制

许多控制系统中采用气缸或液压缸作为驱动元件，并且还需要缸的一系列伸出和缩回动作发生。例如，我们有A和B两个缸，当开始按钮按下时，A缸的活塞向外伸出，而当其完全伸出时，缸B的活塞开始伸出。当活塞均已向外伸出时，A缸的活塞可能需要缩回，当其充分缩回以后，B缸接着再缩回。在对缸的顺序控制的讨论中，经常会给每一个缸A、B、C、D等参考符号。为了说明每一个缸的状态，则会使用“+”号来说明气缸处于伸开状态，使用“-”号说明气缸处于缩回状态。所以上述需要的操作顺序就是A+，B+，A-，B-。图7-20说明了一个可以产生如上顺序的电路图。



177

图7-20 双驱动器顺序操作

操作顺序概述如下：

1）最初，两个缸都已经缩回活塞。按下阀1启动按钮，这将会给阀2施加压力，初始限位开关b-被激活，因此阀3被打开，继而给缸A施加压力，使之活塞向右运动。

2）缸A向右运动，释放限位开关a-。当缸A充分伸出时，限位开关a+打开。这会使阀5打开，并施加压力给阀6使之打开，这样就会给B缸施加压力，引起其活塞向右运动。

3）缸B伸出，释放限位开关b-，当缸B充分伸出时，限位开关b+打开。这会使阀4打

开，并给阀 3 施加压力，继而给缸 A 施加压力，引起其活塞缩回。

4）缸 A 缩回后，释放限位开关 a +，当缸 A 的活塞充分缩回时，限位开关 a - 打开。这将会打开阀 7，使压力施加在阀 6 上，阀 6 打开，这样就会将压力施加于缸 B 上并启动其活塞使之缩回。

5）缸 B 缩回后，释放限位开关 b +。当缸 B 充分缩回时，限位开关 b - 打开，完成循环。

只要按下开始按钮，这个循环可以再次启动。如果我们想要这个系统能够一直运转，那么顺序中的最后一个动作就得引发第一个动作的发生。

实现上述序列的另一种方式是**级联控制**，它涉及用一组阀门来开关控制空气的供给。这种控制可以避免在图 7-20 中，空气被密闭在控制阀的压力管路中，进而发生阻碍阀切换的问题。在级联控制下，操作的顺序可以分成若干组，在每组中，没有一组会出现多于一次以上的缸符号。所以对于 A +，B +，B -，A - 这样的顺序，我们可以分成 A +，B + 和 A -，B - 两组。阀用来在两组之间切换空气的供给。换句话说，空气一会在 A + B + 这组中，然后又被切换到 A - B - 这组中。线路中需要一个启动/停止阀以便选择第一组，如果连续地重复进行这个序列，那么最后要提供一个信号来启动序列反复。每一组的第一个状态是由该组被打开时的电源供给初始化的，后续的动作通过开关阀控制来实现，阀的最后一个动作将初始化下一个被选择的组。图 7-21 显示了该气路图。

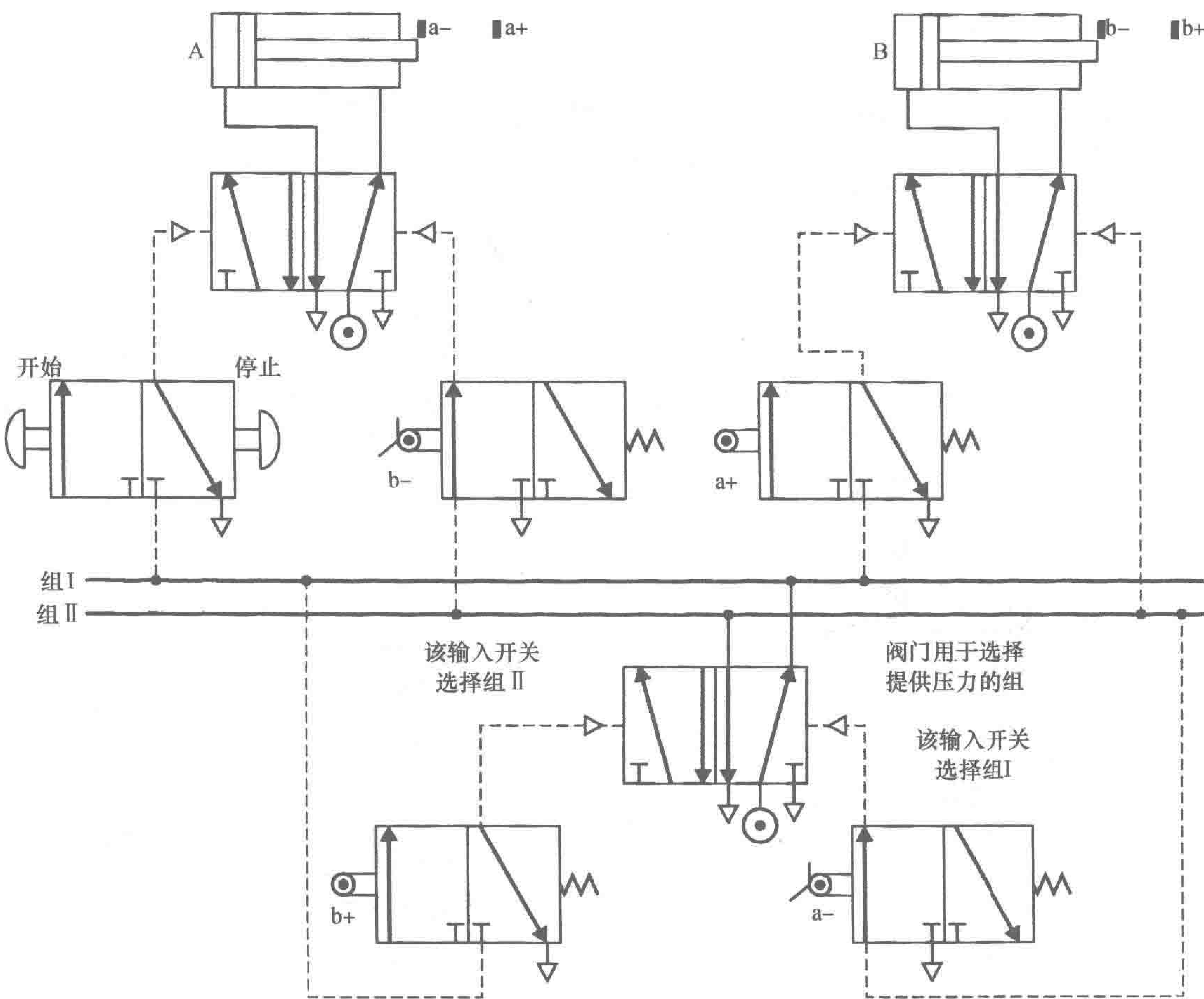


图 7-21　级联控制用于给出 A +，B +，B -，A - 的顺序

7.6 伺服和比例控制阀

伺服和**比例控制阀**都是没有位置限制的控制阀，而且它们的阀芯位移与螺线管电流的大小成比例关系。通常，伺服阀中有一个转矩电动机来移动阀芯（见图 7-22）。改变供给电动机的电流，电枢会发生偏离而带动控制阀芯，进而给出与所供电流相关的流量。伺服阀精度高，但费用也高，一般应用于闭环控制系统中。

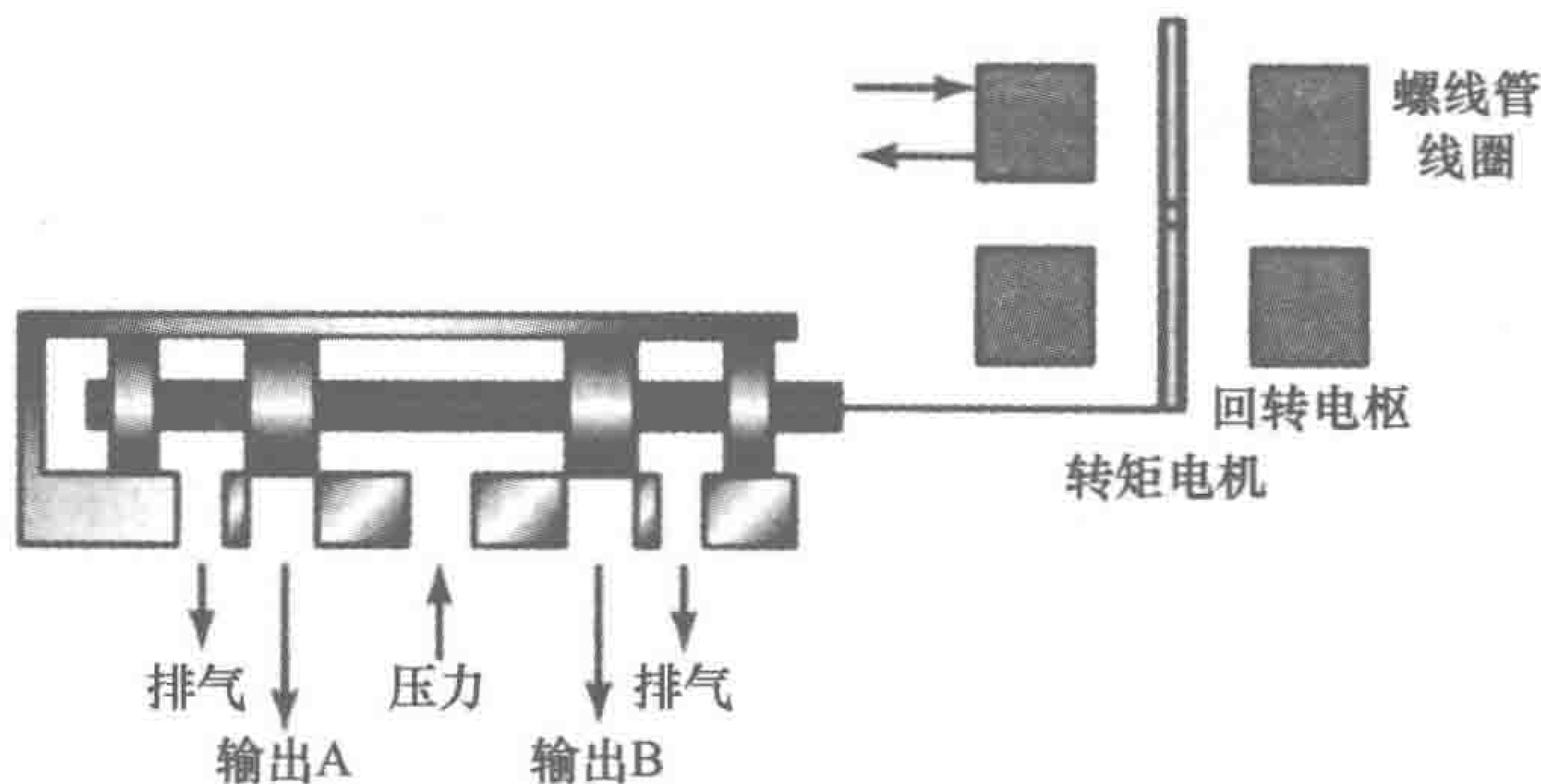


图 7-22 伺服阀的基本形式

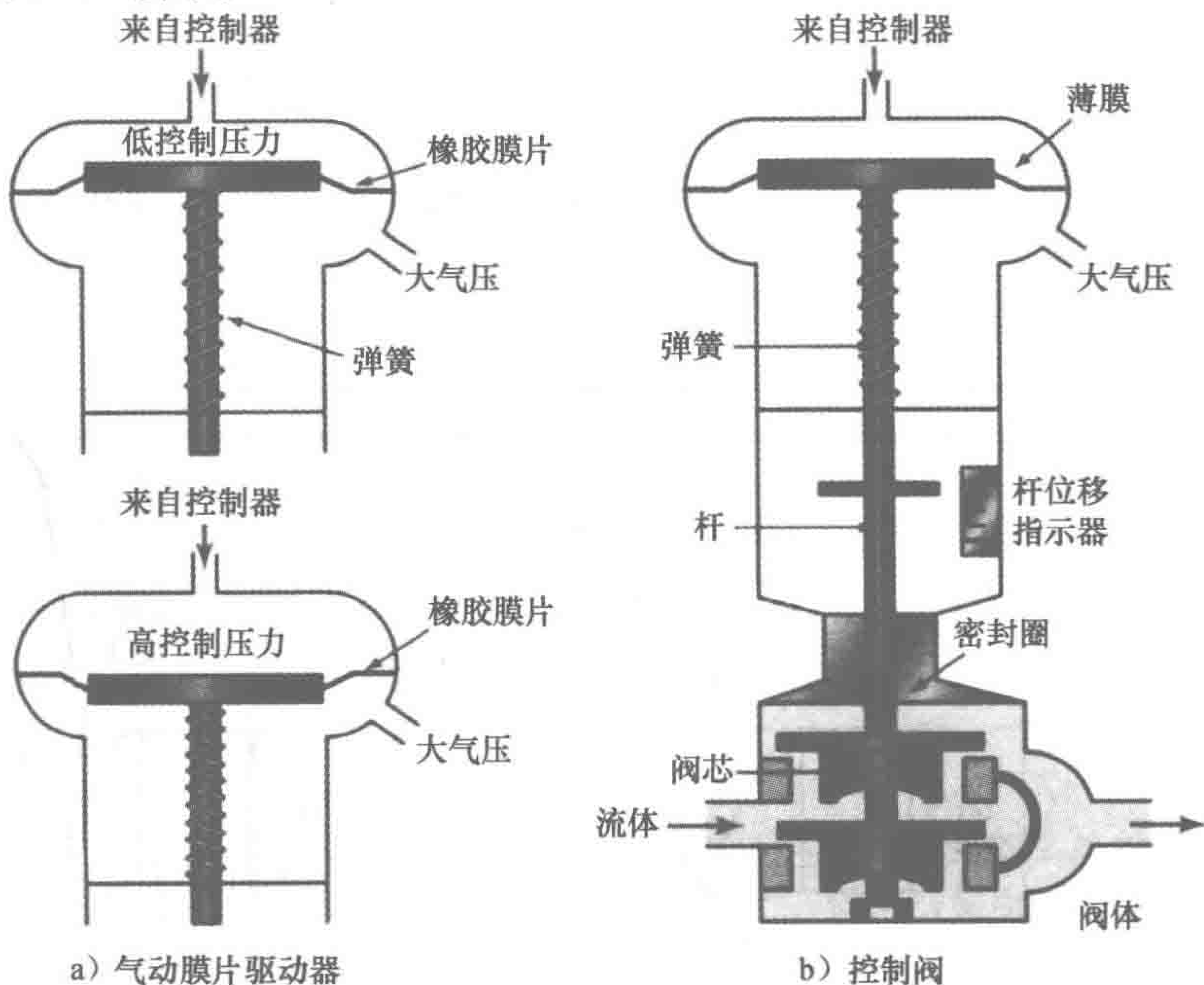
178～179

比例控制阀便宜很多，它的阀芯基本上由阀的螺线圈电流大小直接控制。其一般应用于开环控制系统中。

7.7 过程控制阀

过程控制阀多用于要求流入箱体的液体的流速必须被控制的场合中。这类控制阀的基本工作原理是用一个执行机构移动管路中的活塞，通过改变管路的横截面而使流速改变。

过程控制阀中的气动驱动器最常见的形式是**膜片驱动器**。本质上它是由一个弹性膜片构成，该膜片一面受控制器的输入压强信号的控制，另一面受大气压强的控制，这两种压强差称为**表压**（gauge pressure）。膜片由夹在两个钢盘中间的圆形橡胶做成。来自输入压强的变化会导致膜片中心部分的移动，如图 7-23a 所示。这个移动通过连接到膜片上的轴传递给最终的控制元件，如图 7-23b 所示。



a）气动膜片驱动器　b）控制阀

图 7-23 过程控制阀

作用于轴上的力就是作用于膜片上的力，即表压 P 与膜片面积 A 的乘积。而恢复力则是由弹簧提供的。这样，假设力 F 与弹簧压缩成比例，如果轴移动距离为 x，则有 $F=kx$，其中 k 为常量，又有 $kx=PA$，因此位移 x 与表压成比例。

举例说明：作用于阀上的力是500N，表压是100kPa，问需要多大的膜面积 A 才能使膜片驱动器打开控制阀？作用于膜面积为 A 的力 F 由 $P=F/A$ 提供，因此 $A=500/(100\times1000)=0.005\text{m}^2$。

180

7.7.1　阀体和阀芯

图 7-23b 展示了一个控制流体流量的控制阀的横截面。来自驱动器压力的变化导致膜片的移动，使阀杆移动，结果阀体内的阀芯也发生移动。阀芯限制了流体的流动，它的位置也决定了流量的大小。

有许多形式的阀体和阀芯。图 7-24 列出了几种阀体形式。**单座型**一般用于单通道阀中，因此只需要一个阀芯就可以控制流体了。**双座型**用于将进入阀内的流体分成两个支流（见图 7-23）的阀中，每个支流通过一个阀芯控制的孔。因此在这样的阀中需要两个阀芯。

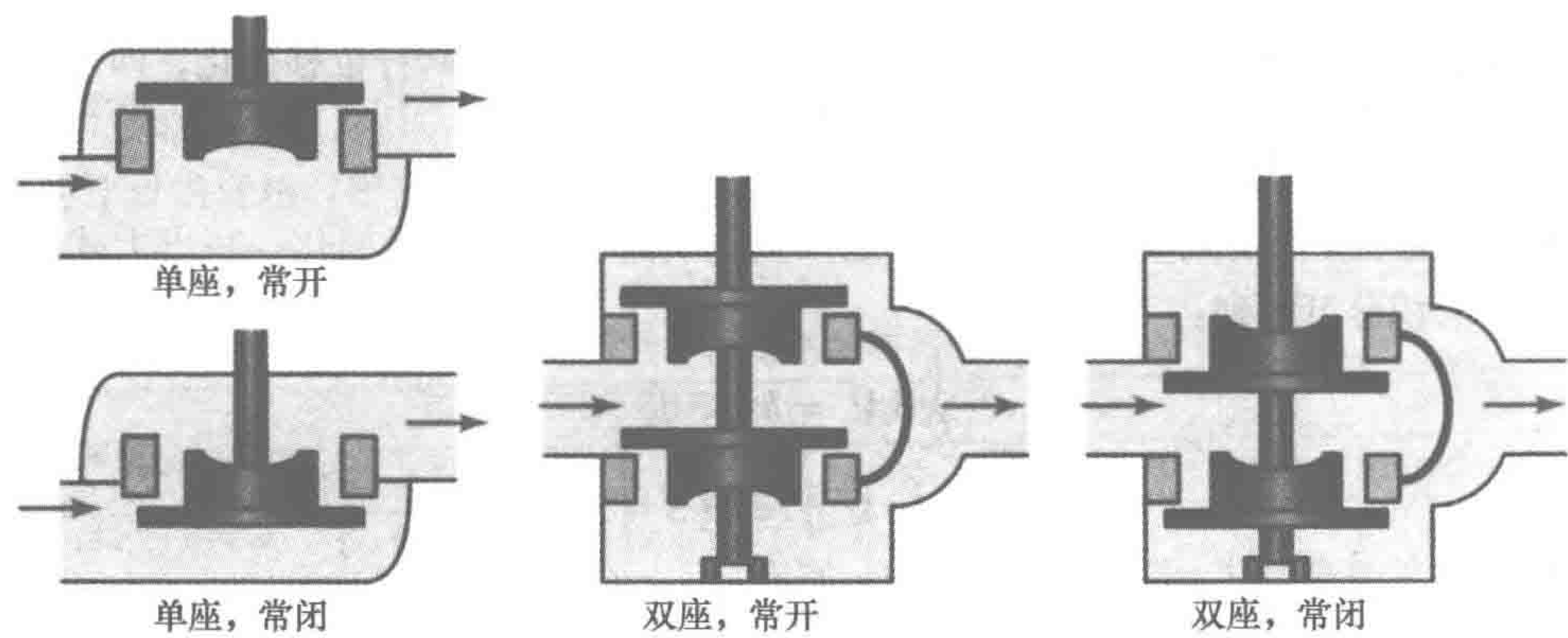


图 7-24　阀体

相比于双座控制阀，单座控制阀的优点是它处于关闭状态时密封性更好，缺点是流体作用于阀芯上的力要大很多，导致执行机构中的膜片必须向阀杆施加相当大的力。这样就会使阀芯的精确定位出现问题。双座型显然在这方面就有了它的优势。阀体的形式也决定了气压的增加是否会导致阀的开或关。

阀芯的形状决定了阀杆的移动与流量之间的关系。图 7-25a 为三种常用的阀芯类型，图 7-25b 为流量百分比和阀杆位移百分比之间的关系曲线。

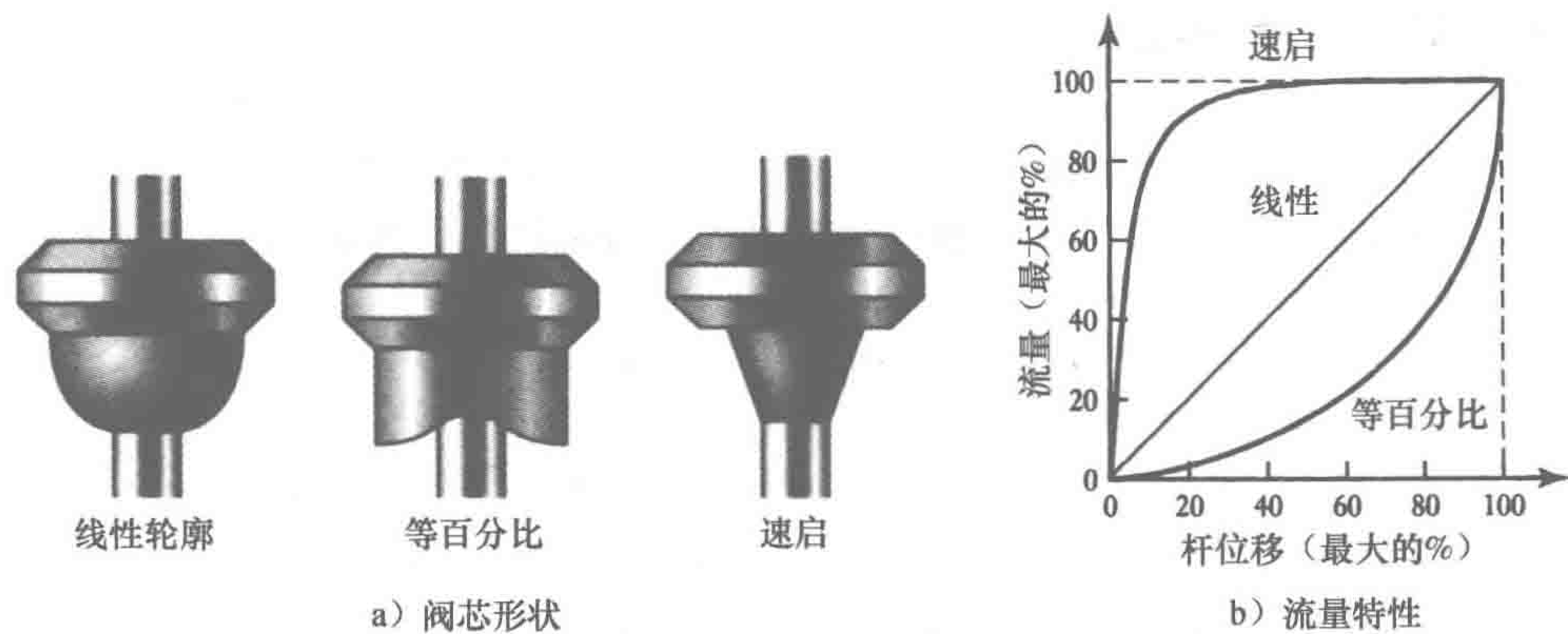


图 7-25　阀芯类型与特性曲线

对于**速启**阀芯，很小的阀杆位移都会造成很大的流量变化。这种类型的阀芯一般用于要求流量开关控制的情况。

对于**线性轮廓**阀芯，流量的变化和阀杆位移变化呈比例关系，即

$$\text{流量的变化量} = k(\text{阀杆位移的变化量})$$

k 为常量。如果Q是阀杆位移为 S 时的流量，Q_{max}是阀杆最大位移为 S_{max}时的最大流量，则有：

$$\frac{Q}{Q_{max}} = \frac{S}{S_{max}}$$

即流量的变化百分比等于阀杆位移的变化百分比。

举例说明：驱动器中阀杆的满量程位移是 30mm，所安装的线性阀塞芯允许的最小流量是 0，最大流量是 $40m^3/s$。当阀杆位移是（a）10mm；（b）20mm 时，控制阀允许通过的流量是多少？由于流量百分比等同于阀杆位移百分比，所以（a）33% 的阀杆位移给出 33% 的流量，即 $13m^3/s$；（b）67% 的阀杆位移给出 67% 的流量，即 $27m^3/s$。

对于**等百分比**型阀芯，流量变化的百分比对应着同等大小的阀杆位移变化，即

$$\frac{\Delta Q}{Q} = k\Delta s$$

其中 ΔQ是流量为Q时的流量变化值，ΔS 是该变化引起的阀芯位置变化。写出上式的微变形式，并对其求积分可得到：

$$\int_{Q_{min}}^{Q} \frac{1}{Q} \mathrm{d}Q = k\int_{S_{min}}^{S} \mathrm{d}S$$

$$\ln Q - \ln Q_{min} = k(S - S_{min})$$

181～182

如果我们考虑流量Q_{max}是由 S_{max}给出的，则有

$$\ln Q_{max} - \ln Q_{min} = k(S_{max} - S_{min})$$

消掉常量 k 后，得到：

$$\frac{\ln Q - \ln Q_{min}}{\ln Q_{max} - \ln Q_{min}} = \frac{S - S_{min}}{S_{max} - S_{min}}$$

$$\ln\frac{Q}{Q_{min}} = \frac{S - S_{min}}{S_{max} - S_{min}} \ln\frac{Q_{max}}{Q_{min}}$$

因此有

$$\frac{Q}{Q_{min}} = \left(\frac{Q_{max}}{Q_{min}}\right)^{(S-S_{min})/(S_{max}-S_{min})}$$

比率$\frac{Q_{max}}{Q_{min}}$称为**可调范围**，记作 R。

举例说明：驱动器中阀杆的满量程位移是 30mm，所安装的等百分比阀芯允许的最小流量是 $2m^2/s$，最大流量是 $24m^2/s$。问当阀杆位移是（a）10mm；（b）20mm 时，控制阀允许通过的流量是多少？用公式

$$\frac{Q}{Q_{min}} = \left(\frac{Q_{max}}{Q_{min}}\right)^{(S-S_{min})/(S_{max}-S_{min})}$$

有（a）$Q = 2 \times (24/2)^{10/30} = 4.6m^3/s$；（b）$Q = 2 \times (24/2)^{20/30} = 10.5m^3/s$。

流量和阀杆位移之间的关系是阀的固有特征。只有在其他管道的压力损失相比于通过阀门本身的压降微不足道时，这个特征才会在实际中出现。如果很大一部分压力损失于管道中，例如只有不足 50% 的压降通过阀门，那阀门的线性特征很可能就变成了速启特征。当需要线性响应并且大部分的压强都在阀门上时，拥有线性特征的阀门会得到广泛应用。对于等百分比

阀，管道中的压降大，会使其更像一个线性特性的阀。基于这个原因，当需要线性响应而却只有很少一部分系统压降通过阀门时，将会用到等百分比阀门。

7.7.2 控制阀的尺寸计算

确定阀体正确尺寸的过程称为控制阀的尺寸计算。阀门全开时经过它的流量Q和阀门大小之间的关系式为：

$$Q = A_V\sqrt{\frac{\Delta P}{\rho}}$$

式中 A_V 是阀门的流量系数，ΔP 是通过阀门的压降，ρ 是流体密度。当写为国际单位时，上式为：

$$Q = 2.37\times 10^{-5}C_V\sqrt{\frac{\Delta P}{\rho}}$$

183

其中 C_V 是阀门的流量系数。或者，它也可以写成：

$$Q = 0.75\times 10^{-6}C_V\sqrt{\frac{\Delta P}{G}}$$

式中 G 为比重或者相对密度。后两种形式的公式源于 US gallon 的原始规范。表 7-1列出了 A_V、C_V 和阀门大小一些典型的值。

表 7-1 流量系数和阀门大小

流量系数	阀门大小（mm）							
	480	640	800	960	1260	1600	1920	2560
C_V	8	14	22	30	50	75	110	200
$A_V\times 10^{-5}$	19	33	52	71	119	178	261	474

为了说明上述情况，考虑确定一个阀门大小的问题，要求该阀能够控制水的流动，使得所需的最大流量为 $0.012\text{m}^3/\text{s}$，并且该流量下阀的允许压降是 300kPa。应用等式

$$Q = A_V\sqrt{\frac{\Delta P}{\rho}}$$

又因为水的密度是 1000kg/m^3，所以有：

$$A_V = Q\sqrt{\frac{\rho}{\Delta P}} = 0.012\sqrt{\frac{1000}{300\times 10^3}} = 69.3\times 10^{-5}$$

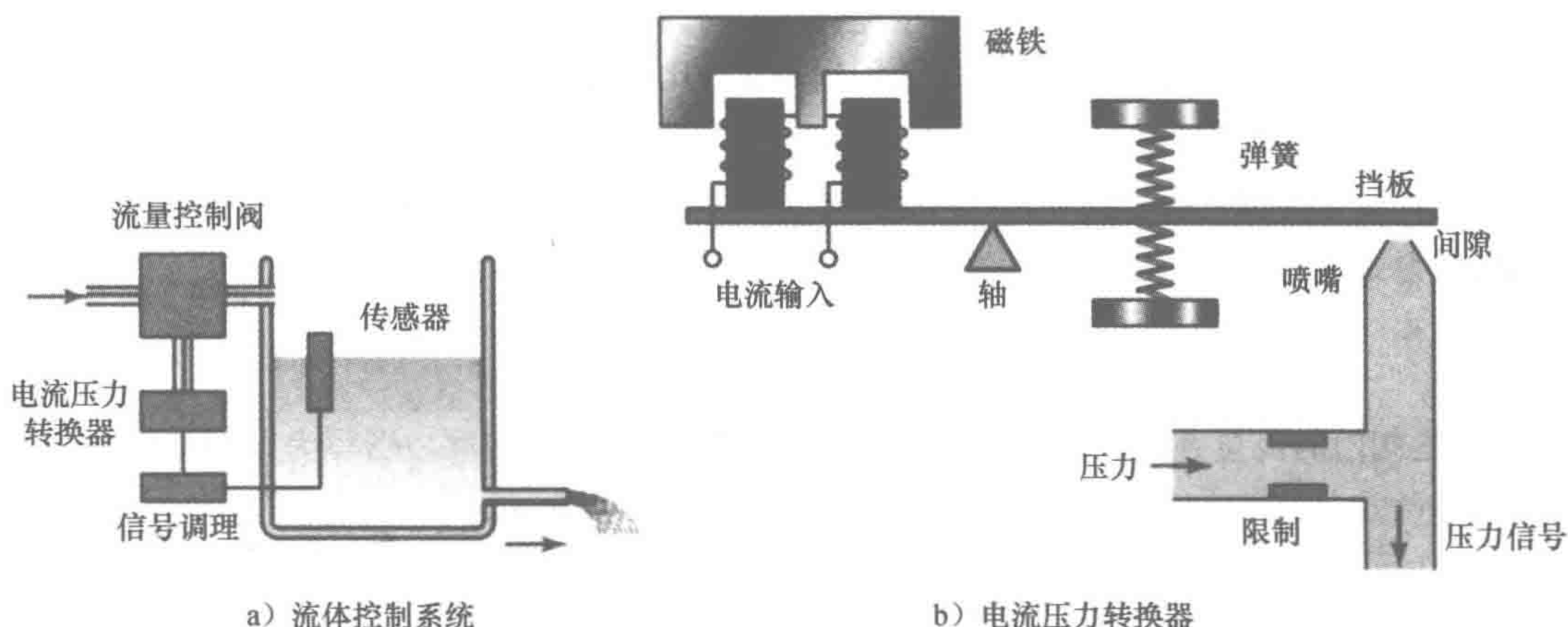
因此，参照表 7-1 可知，阀门大小是 960mm。

7.7.3 流体控制系统举例

图 7-26a 展示了当控制某一变量时系统的基本特征，比如通过控制流入容器的液体的流量来控制液面的高度。液位传感器的输出经过信号调理后，传送到电流压力转换器中作为 4 ~ 20mA的电流信号。经转化得到 20 ~ 100kPa 的表压，启动控制阀控制允许进入容器的流量。

图 7-26b 展示了上述系统中电流压力转换器的基本构成。输入电流通过安装在被磁铁吸引的铁心上的线圈，吸引力的大小取决于所通电流的大小。铁心的移动导致杠杆绕支点转动，接着带动喷嘴上方挡板的移动。挡板和喷嘴之间的位置关系决定着气体从系统中溢出的速度，也即决定着系统的气压。挡板上的弹簧用来调节转换器的灵敏度，以便 4 ~ 20mA 的电流可以产生 20 ~ 100kPa 的表压。这些是在这类系统中常用的标准值。

184

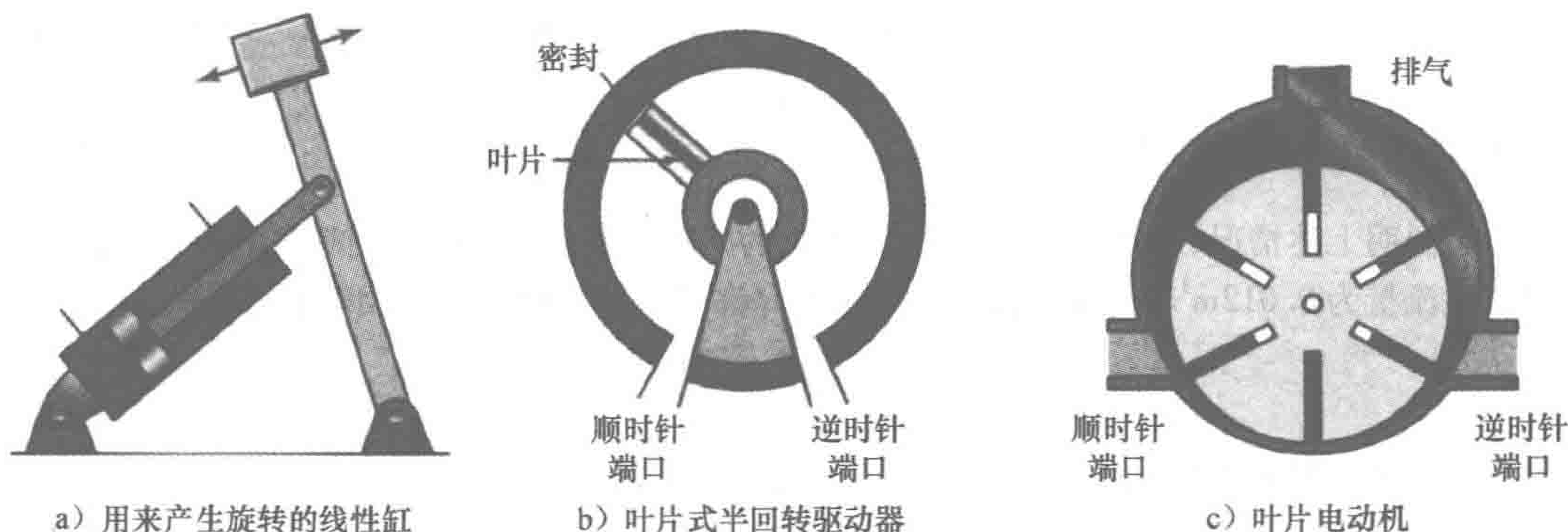


a）流体控制系统　b）电流压力转换器

图 7-26　例子

7.8　旋转驱动器

用合适的机械连杆，一个直线缸可以用于产生小于一周的圆周运动，图 7-27a 为类似装置的原理。另一种形式是**叶片式半回转驱动器**（见图 7-27b）。两端口之间的压力差将导致叶片的旋转，并带动轴的旋转（转速是压力差的度量）。根据压力差的不同，叶片可以顺时针转动也可以逆时针转动。



a）用来产生旋转的线性缸　b）叶片式半回转驱动器　c）叶片电动机

图 7-27　旋转驱动器

为实现大于一周的旋转运动，可采用气动电动机，其中的一种形式就是叶片电动机（见图 7-27c）。在一个有插槽的偏心转子中，叶片将被迫紧贴着缸体的壁旋转。叶片把整个空间分成几个单独的隔间，每个隔间的大小都从入口到出口逐渐变大。进入隔间的气体作用于叶片上，然后引起转子的旋转。通过不同的入气口可以使电动机朝反方向旋转。

185

本章小结

气动系统用的介质是空气，**液压系统**用的介质是油。气动系统的主要缺点是空气的可压缩性。液压系统可以用于更高的功率控制装置，但比气动系统更昂贵，并且液压油泄漏会带来一定危害，相比之下空气泄漏就不会带来这个问题。

气动和液压系统使用**方向控制阀**引导流体通过整个系统。这种阀是开/关阀。这种阀的符号是表示其开关位置的正方形，正方形中的符号表示开关位置被激活时的连接。

液压缸或气缸包括一个圆筒形缸管和可沿着缸管滑动的活塞/柱塞。有两种基本类型，**单作用缸和双作用缸**。单作用缸控制压力只被施加到活塞的一侧，另一侧与大气连通，弹簧通常用来提供活塞的反向移动。双作用缸是指控制压力被施加到活塞的每一侧时的情况。

伺服和比例控制阀都是无限位置阀，阀芯的位移与通过螺线管的电流成比例。

过程控制阀用于控制流体的流量。这类控制阀的基本工作原理是，用一个驱动器移动管中的阀芯，通过改变管的横截面而使流量得到控制。阀体和阀芯的种类比较多，不同的种类决定了不同的阀控制流体流动的方式。

习题

7.1　请详细描述（a）提升阀；（b）梭阀（双单向阀）。

7.2　解释先导阀的原理。

7.3　解释为什么顺序阀只有当一个操作完成后才能启动另一个操作。

7.4　绘制以下阀的符号：（a）卸压阀；（b）2/2 阀，该阀有一个按钮和一个弹簧的驱动器；（c）4/2 阀；（d）单向阀。

7.5　阐述当开始按钮被按下时，图 7-28 中气缸 A、B 所产生的操作顺序。其中 a－、a＋和 b－、b＋是用来检测气缸完全缩回以及完全伸出的限位开关。

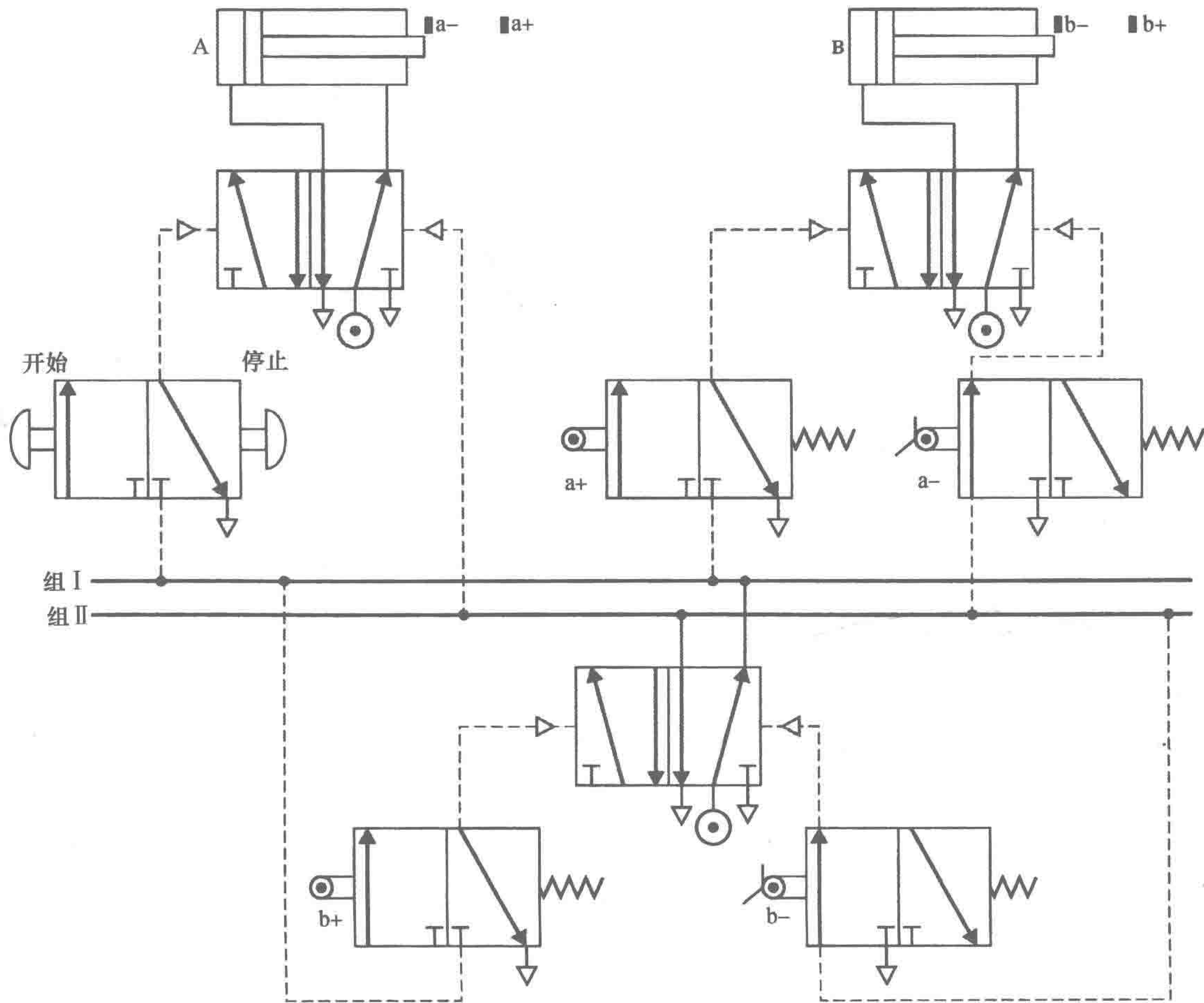


图 7-28　习题 7.5

7.6 设计一个气路图，控制序列为：首先是 A+，接着是 B+，然后 A-和 B-同时动作。

7.7 打开一个过程控制阀需要 400N 的力。用膜片式驱动器打开阀的表压为 70kPa，需要多大的膜片面积来实现？

7.8 气动系统在 1000kPa 的压力下工作。移动一个 12kN 力的负载需要多大直径的气缸？

7.9 一个液压缸用于在 10s 内移动工件 50mm。移动工件需要 10kN 的力。如果所用的缸的活塞直径为 100mm，确定所需要的工作压力和液压油的流量。

7.10 一个驱动器阀杆运动的整个行程为 40mm。安装在带有线性阀芯的过程控制阀上，该阀具有最小流速 0 和最大流速 $0.2m^3/s$。当阀杆运动的行程为：（a）10mm；（b）20mm 时，流量为多少？

7.11 一个驱动器阀杆运动的整个行程为 40mm。安装在带有等百分比阀芯的过程控制阀上，该阀具有最小流速 $0.20m^3/s$ 和最大流速为 $4.0m^3/s$。当阀杆运动的行程为：（a）10mm；（b）20mm 时，流量为多少？

186～187

7.12 需要什么样的过程控制阀尺寸才能在最大流量为 $0.002m^3/s$，阀的允许压降为 100kPa 的条件下控制水的流量？水的密度为 $1000kg/m^3$。

第8章 机械驱动系统

本章目的

通过本章的学习，读者将能够：

- 为平动－转动、转动－转动、转动－平动和往复运动等运动转换形式选择可能的机械驱动系统。
- 为执行系统评估连杆机构、凸轮、齿轮、棘轮棘爪、带（链）传动和轴承的性能。

8.1 机械系统

机构可以看作运动的转换装置，通过机构可以将运动从一种形式向其他形式转换。例如，机构可以把一个直线运动转换为转动，可以把沿着一个方向的直线运动转换成与原方向成直角的直线运动，可以把直线往复运动转换成转动。在内燃机里面，活塞的往复运动可以转换为曲柄的转动并带动传动轴。

机构单元包括连杆机构、凸轮机构、齿轮机构、齿轮齿条机构、带（链）传动等。例如，齿轮齿条可以把转动转换成直线运动，平行轴间的齿轮可以降低轴的转速。锥齿轮可以把转动的转轴改变90°。带（链）传动可以将围绕一个轴的转动转换成围绕另一个轴的转动。凸轮和连杆可以让运动以一个特定的方式改变。本章将讨论这些机构的基本特点。

早在使用微处理器的机电一体化系统出现之前，上述机构就已经被用于类似的控制场合。比如转轴和凸轮曾经用在家用洗衣机上，它可以给出动作时序，如打开阀门让水进入、停止进水、启动加热器等。而现在的洗衣机大多数已经使用微处理器来给出动作时序。再比如，由游丝摆轮、齿轮机构带动指针转动的手表，现在很大程度上已经被集成电路和液晶显示的手表替代了。机电一体化方法往往更简洁、花费更少。

然而直到现在，机构仍然在机电一体化系统中占有一席之地。比如，全自动相机的机电一体化系统就是由某种机构来调节光圈大小，从而获得正确的曝光。

即使电子设备现在已经广泛应用于那些当年充满着各种机械结构的设备中，但是机构仍然可以为我们提供这些功能：

1）力放大，比如杠杆。

2）改变速度，比如齿轮。

3）改变传动轴，比如同步带。

4）某些特殊类型的运动，比如急回机构。

运动学只研究机构的运动，而不考虑力的作用。当只研究运动而不考虑其中的力和能量时，就是运动学分析。本章将介绍运动学。

8.2 运动类型

刚体运动可以看作转动和平动的结合，在三维空间里，平动可以看作沿着三条坐标轴中的一条或几条轴进行分解得到的直线运动分量（见图8-1a），旋动可以看作绕着三条坐标轴中的

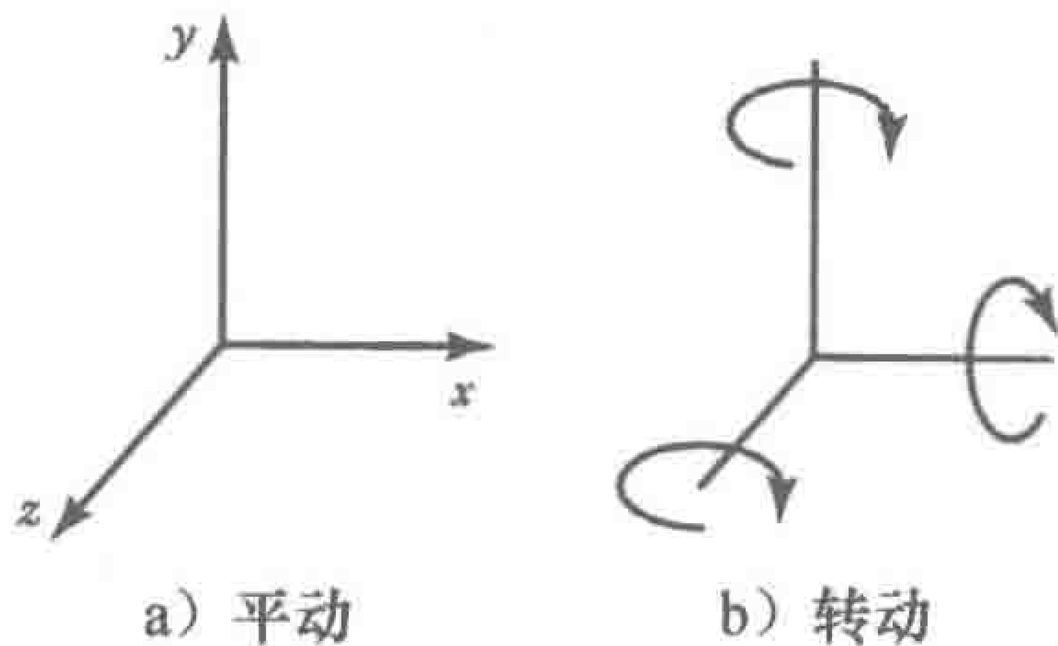
一条或几条进行转动的分量（见图 8-1b）。

复杂运动可以看作平动和转动的结合。比如你从桌子上捡起一支铅笔的动作，可能包括手以一定角度接近桌子的运动、手旋转的动作，以及张开手指并移动到相应位置去抓住铅笔的动作。这其实就是一系列相当复杂的运动。可是，我们完全能够把它分解成平动和转动。这样的分析对于人去捡铅笔来说可能没什么作用，但是对于控制机器人完成该任务就显得意义重大。为了设计出一系列机构来完成不同的动作，就必须对复杂运动进行分解。例如，发送给机构的一系列控制信号可能就是这样的：关节 1 旋转 20°，连杆 2 平移 4mm 等。

189

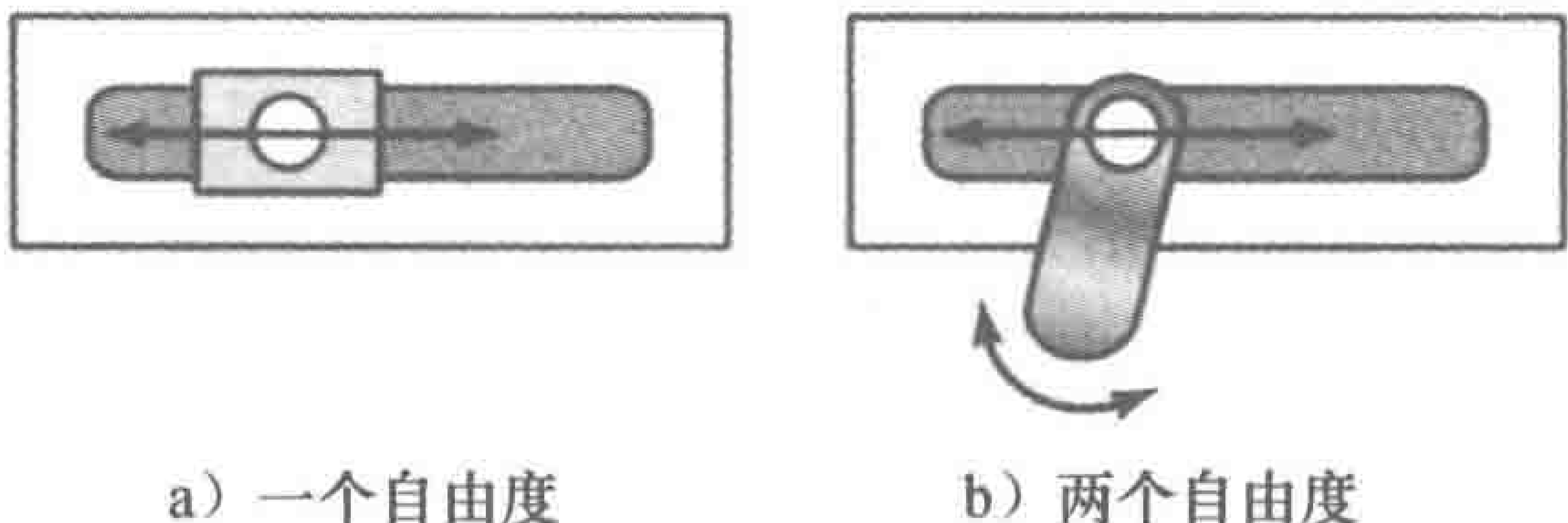
8.2.1　自由度和约束

机械元件设计中很重要的一方面是机械元件的方位和排列。空间中的一个自由物体可以沿着三个独立的相互垂直的方向直线运动，也可以绕着它们转动（见图 8-1），我们称该物体具有 6 个自由度。**自由度**是为了产生运动所必需的运动分量的数目。如果限制一个关节只能沿一条直线运动，那么它的平动自由度就减少到 11。图 8-2a 就是这样一个只有一个平动自由度的关节。如果一个关节被限制在一个平面上移动，那么它就是具有两个平动自由度。图 8-2b 中的关节具有一个平动自由度和一个转动自由度。



a）平动　b）转动

图 8-1　运动类型



a）一个自由度　b）两个自由度

图 8-2　关节

设计中的问题是经常减少自由度的数目，这就需要找到合适的约束数目和方位。如果没有约束，那么一个物体就具有 6 个自由度，减少一个自由度，就需要一个约束。如果没有冗余约束（又称为虚约束），那么自由度数就用 6 减去约束数目。然而，冗余约束经常出现，因此对于单个刚体，有这样一个基本规则：

6 - 约束数目 = 自由度数 - 冗余约束数

因此，如果想要固定一个刚体，也就是说，让它的自由度为 0，那么若没有冗余约束，则需要 6 个约束。

在设计中经常提到一个概念是“**最少约束原则**”。这个原则告诉我们，如果要固定一个物体，或者要让一个物体按照某个固定方式运动，那么应该找到需要的约束的最小的数目，也就是说，不应该存在冗余约束。这也经常用于**运动学设计**中。

举个例子，我们要让一个转轴仅仅绕着一个轴转动，而不会发生直线运动，那么我们就要把自由度减少到 1 个，因此至少需要 5 个约束。任何多余的约束都会造成冗余。至于固定方式，可以在轴的一端安装一个球轴承，另一端安装一个滚柱轴承（见图 8-3）。这对轴承共同限制了沿转轴垂直方向的 y 轴的平动和绕着 z、y 轴的转动。球轴承则限制了转轴沿着 x、z 轴的平动。因此这里共有 5 个约束，也就是只留下了一个自由度，即绕着 x 轴的转动。如果两边安装的都是球轴承，那么这两个轴承就都

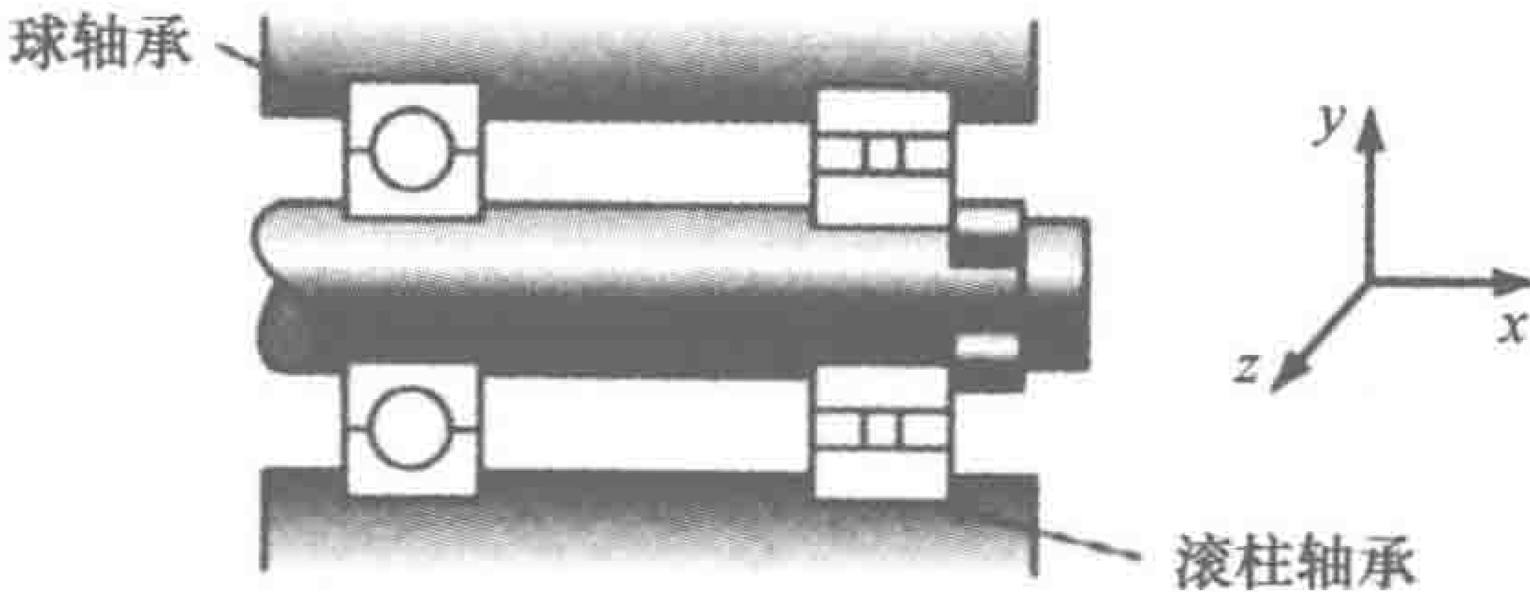


图 8-3　没有多余自由度的轴

190

起到了限制沿着 x、z 轴平动的作用，于是形成了冗余约束。这种冗余可能造成损坏。如果两边安装的都是球轴承，为了不产生冗余，其中一个轴承的外圈就不能固定在壳体上，这样在轴向上就可以产生一定程度的滑动。

8.2.2　负载

机构就是用来支承和传递负载的结构，因此对单个元件承载能力的分析是必要的。那么就要考虑机构的尺寸问题，比如说，使其在特定负载下具有足够的强度和刚度。

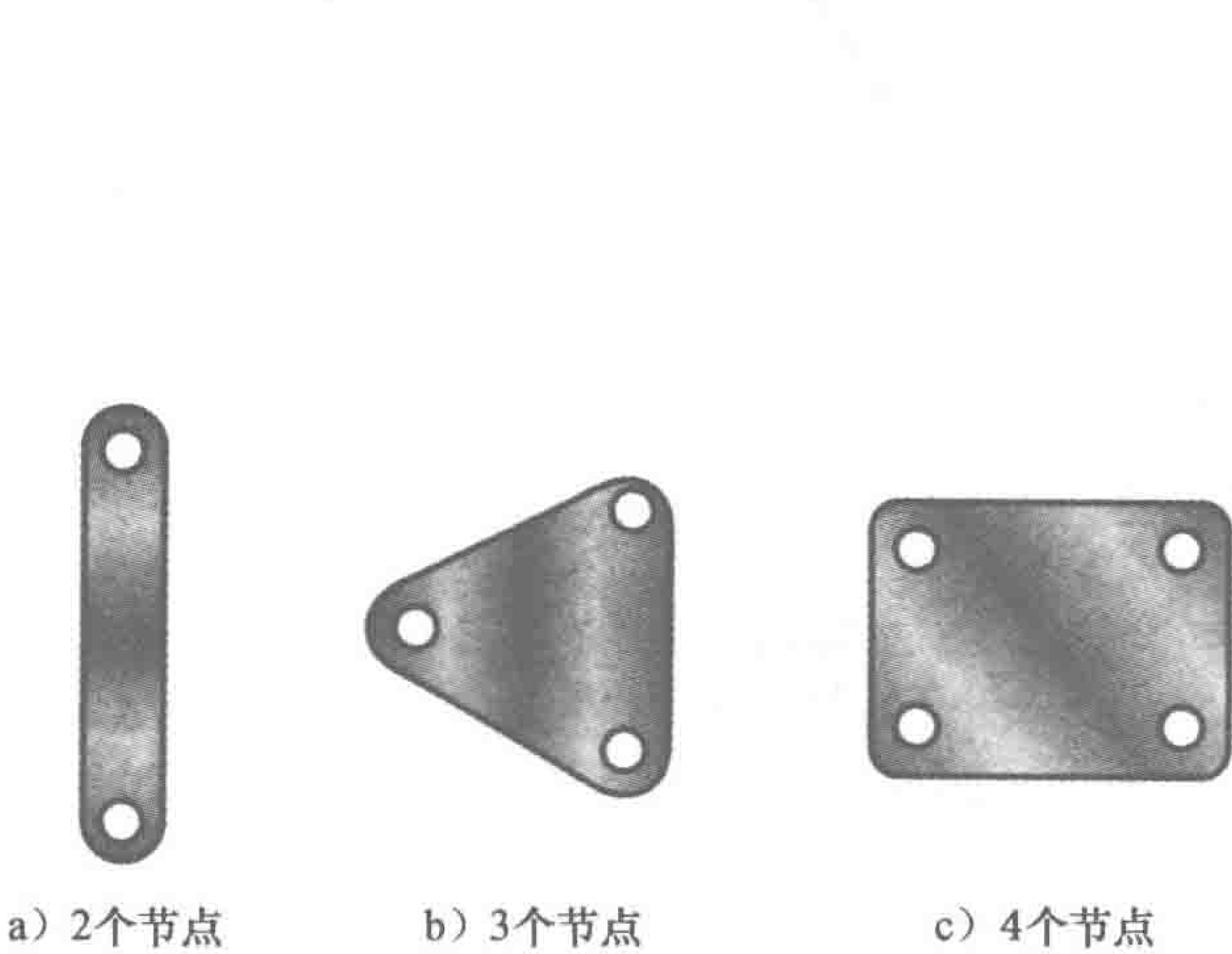
8.3　运动链

当我们仅考虑运动而不考虑机构的受力时，我们可以把机构看成一系列连接的组合。机构中每个与其他零件发生相对运动的零件，都称为**连杆**。连杆不一定是刚体，但一定要耐冲击，从而可以忽略在传递力的过程中的变形。因此它经常可以用刚体来表示，其上有两个或多个点，用于连接其他的连杆，这些点称为**节点**。每个连杆都可以与旁边的连杆发生相对运动，图 8-4 就是分别具有 2 个、3 个、4 个节点的连杆。把几个连杆在节点处连接起来，就是**关节**它允许相连的连杆之间产生运动。杠杆、曲柄和活塞、滑块、滑轮、传送带和转轴都是连杆的例子。

191

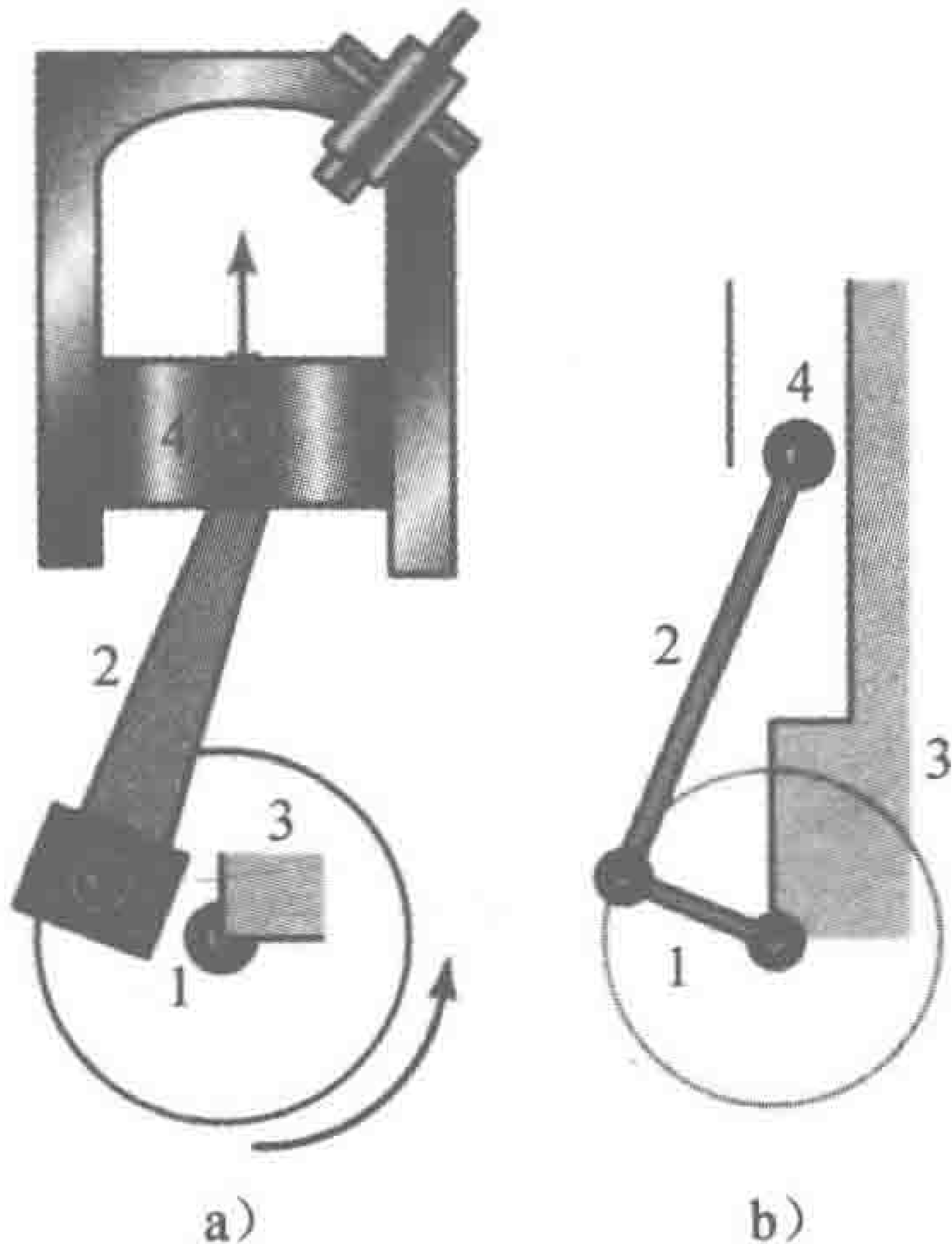
一系列的关节和连杆称为**运动链**。运动链传递运动，必须有一个连杆是固定的。从一个连杆的运动可以预测其他连杆的相关运动。通过固定不同的连杆，可以从一个运动链中获得不同的机构。

考虑电动机汽车发动机，活塞的往复运动可转换为安装在固定框架轴承上的曲轴的转动（见图 8-5a），我们可以把这个机构表示成四个连杆（见图 8-5b）。连杆 1 是曲轴，2 是连接杆，3 是基座，4 是滑块，即活塞，活塞相对于基座活动（详见 8.3.2 节）。



a）2个节点　　b）3个节点　　c）4个节点

图 8-4　连杆



a）　　b）

图 8-5　简单发动机机构

很多机构的设计是基于两种基本运动链形式——四杆链和滑块曲柄链。下面介绍这两种链可能的形式。

8.3.1　四杆链

四杆链就是四根连杆通过 4 个关节连接在一起，可以绕关节转动。图 8-6 就是改变各连杆之间长度关系后产生的几种形式。如果最短杆和最长杆的长度之和小于或等于另外两根杆之和，那么至少一根杆可以相对于固定连杆绕一整圈，否则所有的连杆都不能转一整圈。这就是

192

格拉晓夫条件（Grashof condition）。图 8-6a 中，连杆 3 是固定的，该杆长使 1 和 4 可以摇摆但是不能旋转。这就是**双杆机构**。相对于连杆 1 缩短连杆 4 之后，连杆 4 可以整周旋转（见图 8-6b），这是**曲柄摇杆机构**。连杆 1 和连杆 4 一样长且都能旋转，那么这就是一个**双曲柄机构**（见图 8-6c）。通过改变固定的连杆，就可以得到其他形式的机构。

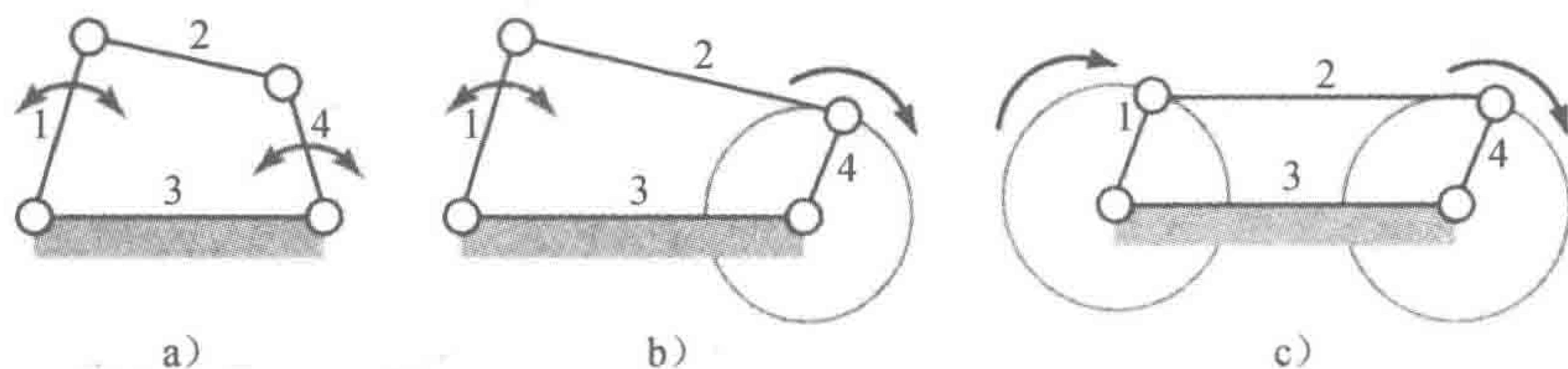


图 8-6　四杆链的例子

图 8-7 说明了如何用这样的机构给电影摄影机进片。如果连杆 1 旋转，使连杆 2 末端锁入胶片的链齿，然后把它往前拉，然后放开，抬起来，再继续锁入下一个链齿。

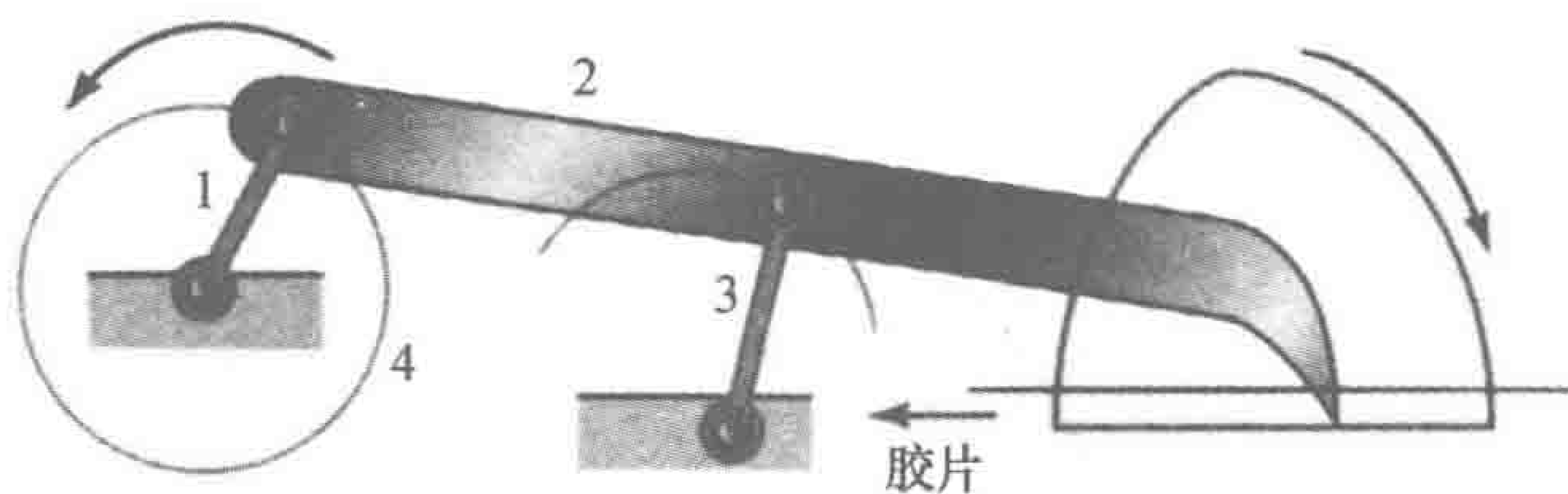


图 8-7　电影胶片推进机构

有些连杆机构有**肘节位置**，在这些位置，连杆机构对于节点两边任意一根连杆的输入都没有任何反应。图 8-8 中就是这样一个肘节机构，它用在卡车货箱的门上，当连杆 2 达到水平位置的时候，无论再受到怎样的负荷，连杆 2 都不会再运动。该连杆机构还有另外一个肘节位置，且当连杆 3 和连杆 4 都垂直时，后挡板也是垂直的。

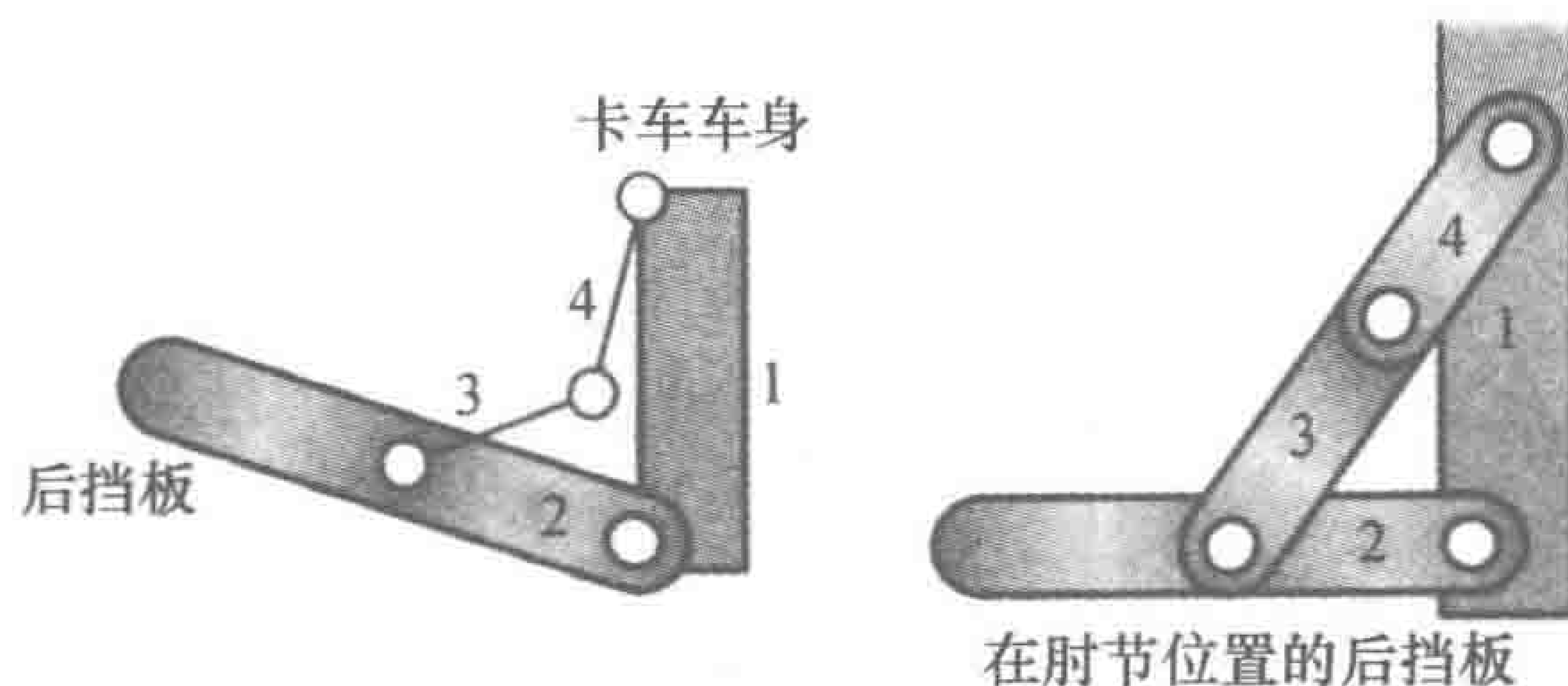


图 8-8　肘节连杆机构

8.3.2　滑块曲柄机构

这种机构包含一个曲柄、一个连接杆和一个滑块，这实际上就是图 8-5 发动机中的那种简单机构。图 8-5 中，连杆 3 是固定的，也就是说曲柄的旋转中心和活塞所在槽之间没有相对运动。连杆 1 是旋转的曲柄，连杆 2 是连接杆，连杆 4 是相对于固定连杆运动的滑块（活塞）。当活塞往复运动时，即连杆 4 往复运动时，曲柄 1 就会强制旋转。因此这种机构把往复运动转换成了转动。图 8-9 显示了这种机构的几种状态。要观察机构是如何运转的，一种很有用的办法是制作一个硬纸板的轮廓，然后移动连杆，观察运动。你还可以改变连杆的长度，看看会产生什么样的变化。

193

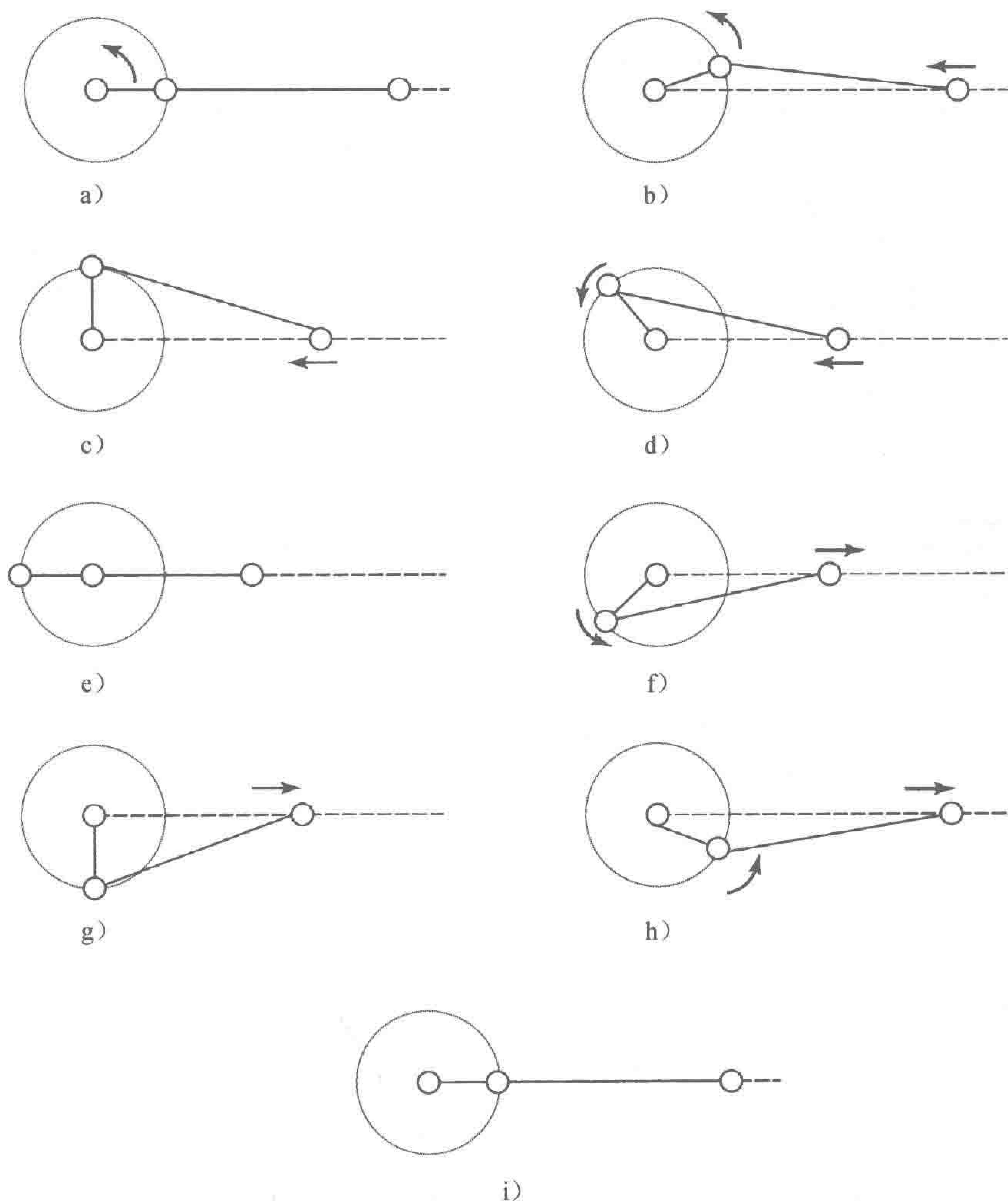


图 8-9　滑块曲柄机构中连杆的位置序列

图 8-10 展示的是**急回机构**。它包括一个绕固定轴转动的曲柄 AB，一根摆动杆 CD。当 AB 转动的时候，滑块在 B 点沿着 CD 滑动，于是 CD 就绕 C 摆动起来，连杆 DE 则带着 E 前后运动。E 可以作为一台机器的柱塞，上面可以连接一个切割装置。当 AB 到达图中 AB_1 和 AB_2 的位置时，E 到达它的极限位置。当 AB 逆时针从 B_1 到 B_2 时，E 完成了整个行程，即切割冲程。当 AB 继续逆时针从 B_2 到 B_1 旋转时，E 又完成了一个反向的完整冲程，即回冲程。由于 AB 的旋转是匀速的，而逆时针从 B_1 到 B_2 的旋转角度大于从 B_2 到 B_1 的旋转角度，所以切割冲程的时间多于返回冲程，因此该机构称为急回机构。类似地，也可以像图 8-9 那样制作一个模型。

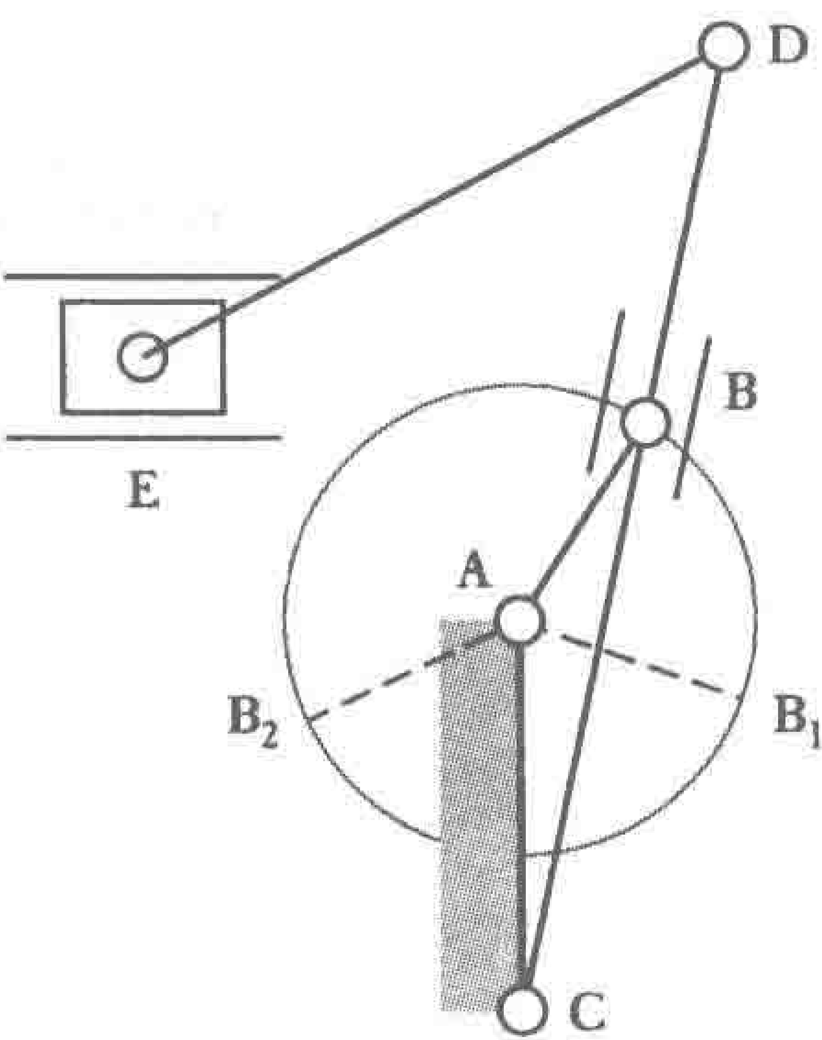


图 8-10　急回机构

8.4　凸轮

凸轮可以通过自身的旋转或摆动往复运动或摆动传递给另一个**从动件**（见图 8-11）。随着凸轮的转动，从动件跟着上升、停留、下降，在各部分停留的时间取决于凸轮的形状。上升

期、下降期、停留期指的是凸轮使从动件上升、下降、停留在同一高度的时间，前两个取决于凸轮的轮廓，第三个发生在凸轮半径不发生改变的圆形部分。

根据凸轮不同的形状和从动件类型的不同，可以产生不同的运动。图 8-12 就是一个偏心轮和尖形从动件的组合，右边是从动件相对于偏心轮转角的位移曲线。这是个圆形的凸轮，只是它的旋转中心并不是圆心。它使从动件产生简谐振动，常常用于泵中。凸轮旋转中心到凸轮从动件接触点之间的距离就是凸轮和从动件的相对距离。图 8-12 告诉我们从动件位移是如何随着凸轮的转动而变化的。该垂直位移图是通过使凸轮旋转不同角度，然后将接触点的位置等高地投影在右侧的图中来获得的。

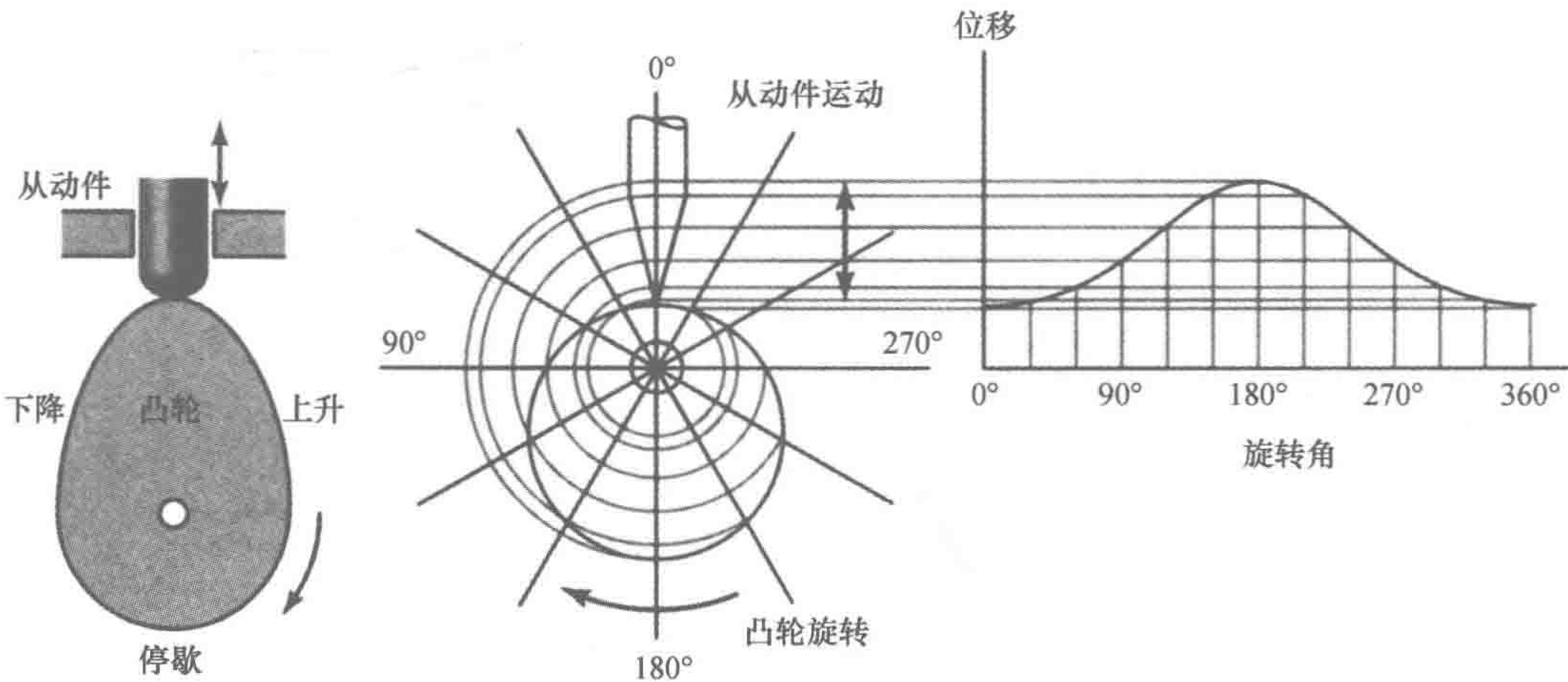
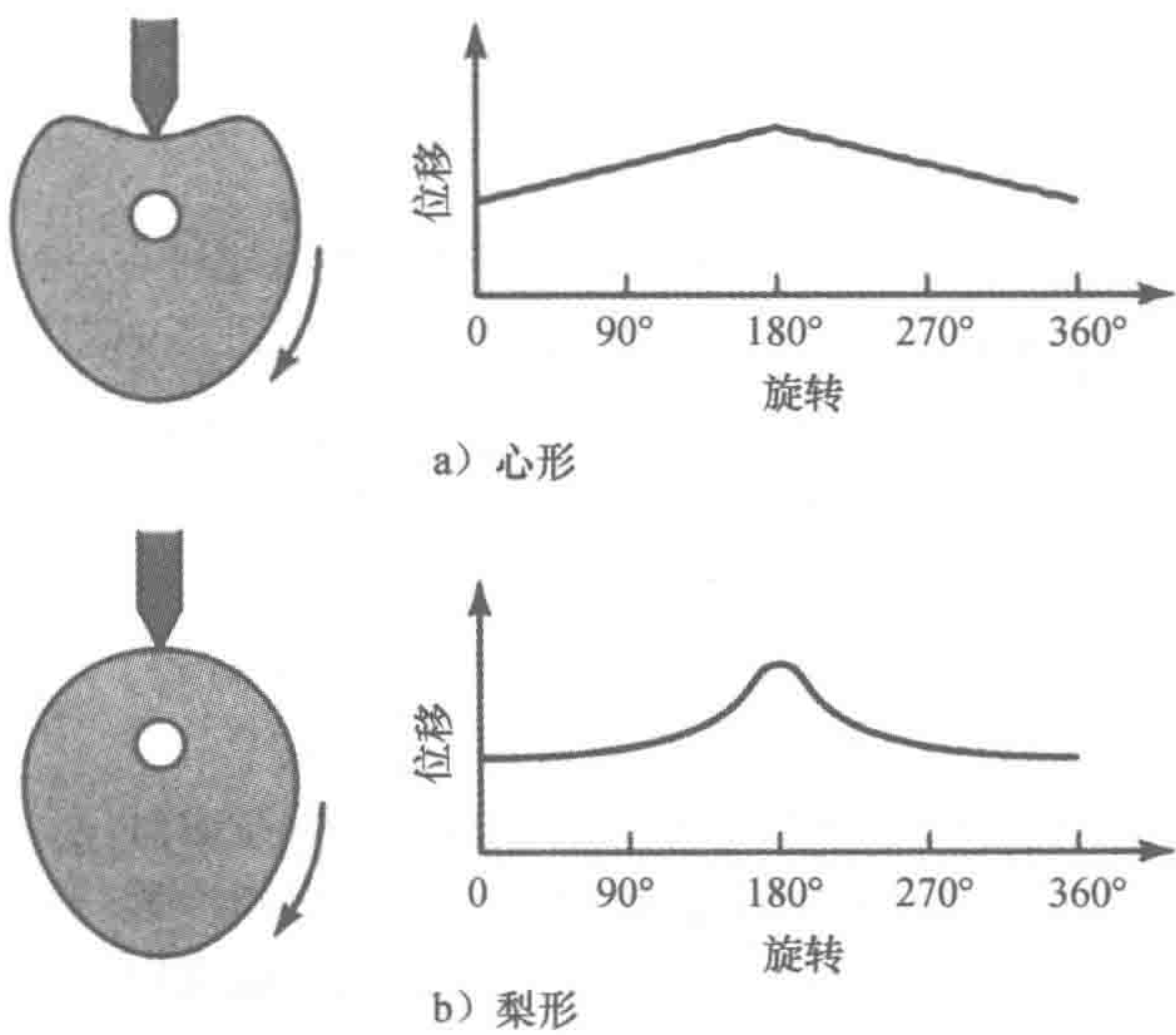


图 8-11 凸轮和凸轮从动件

图 8-12 偏心轮位移图

图 8-13 是使用另外两种形状的凸轮和点或刀从动件形成的位移曲线。



a）心形

b）梨形

图 8-13 凸轮

图 8-13a 中的心形凸轮使从动件的位移以相同的速度上升或下降。图 8-13b 中的梨形凸轮的从动件位移大约一半的行程是平稳的，其余的 1/4 部分呈现出对称的上升和下降。这种机构常用于发动机的阀门控制。停歇的时候阀门打开，这期间油气混合物可以进入

缸体。停歇时间越长，即半径相等的凸轮表面的长度越大，允许缸体充入可燃气体的时间就越长。

图 8-14 是一些不同形状凸轮和从动件的例子。滚动从动件主要是球轴承或者滚柱轴承。这类机构的优点是摩擦力小，但是更贵一些。平面从动件也经常使用，因为它们更便宜而且比滚动从动件更小。这些从动件多用于发动机阀门凸轮。由于凸轮在使用中可能变得干燥，因此常和润滑系统一起使用或者直接浸在油中。

195

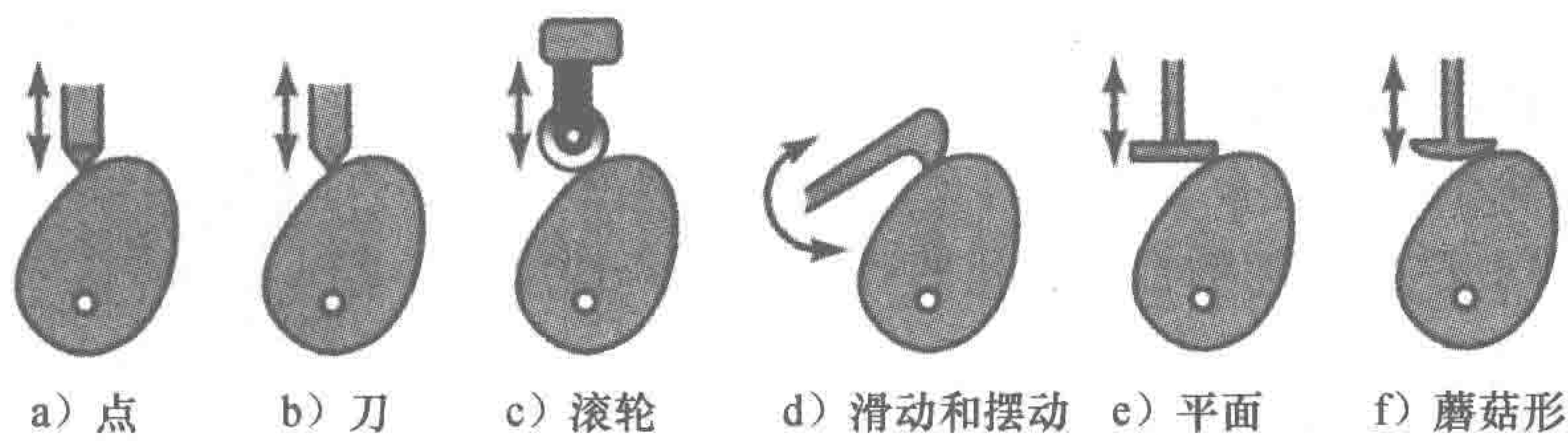


图 8-14　凸轮从动件

8.5　齿轮

齿轮系是广泛用于传递或转换旋转运动的一种机构。它们常用于需要改变转速或转矩的情况。比如汽车的变速箱能使司机根据地形需要的速度、转矩和发动机的功率进行匹配。

齿轮可以用于平行轴（见图 8-15a）和成角度轴（图 8-15b）之间传递旋转运动，**锥齿轮**用于轴线相交的情况，见图 8-15b。当两齿轮互相啮合时，比较大的那个叫作**直齿轮**或者**冠状齿轮**，比较小的那个叫作**小齿轮**。用于平行轴之间传递运动的齿轮可能有着与轴线平行的齿面，这种叫作**直齿轮**（见图 8-15c）。还有可以换成带有斜角度齿形的齿轮，这种叫作**斜齿轮**（见图 8-15d）。这种齿轮的优点是齿间可以逐步啮合，于是运行更平滑而且寿命更长。然而，正是因为有了这个相对于转轴轴线的角度，会产生一个轴向力分量，施加在轴承上。但是我们可以通过**双斜齿轮**（见图 8-15e）来克服它。

196

图 8-16 是两个相互啮合的齿轮 A 和 B，如果 B 有 80 齿，A 有 40 齿，那么相同时间内 A 转 2 圈，则 B 只能转 1 圈。因此 A 的角速度 ω_A 就是 B 的角速度 ω_B 的 2 倍，也就是：

$$\frac{\omega_A}{\omega_B}=\frac{\text{B 齿数}}{\text{A 齿数}}=\frac{80}{40}=2$$

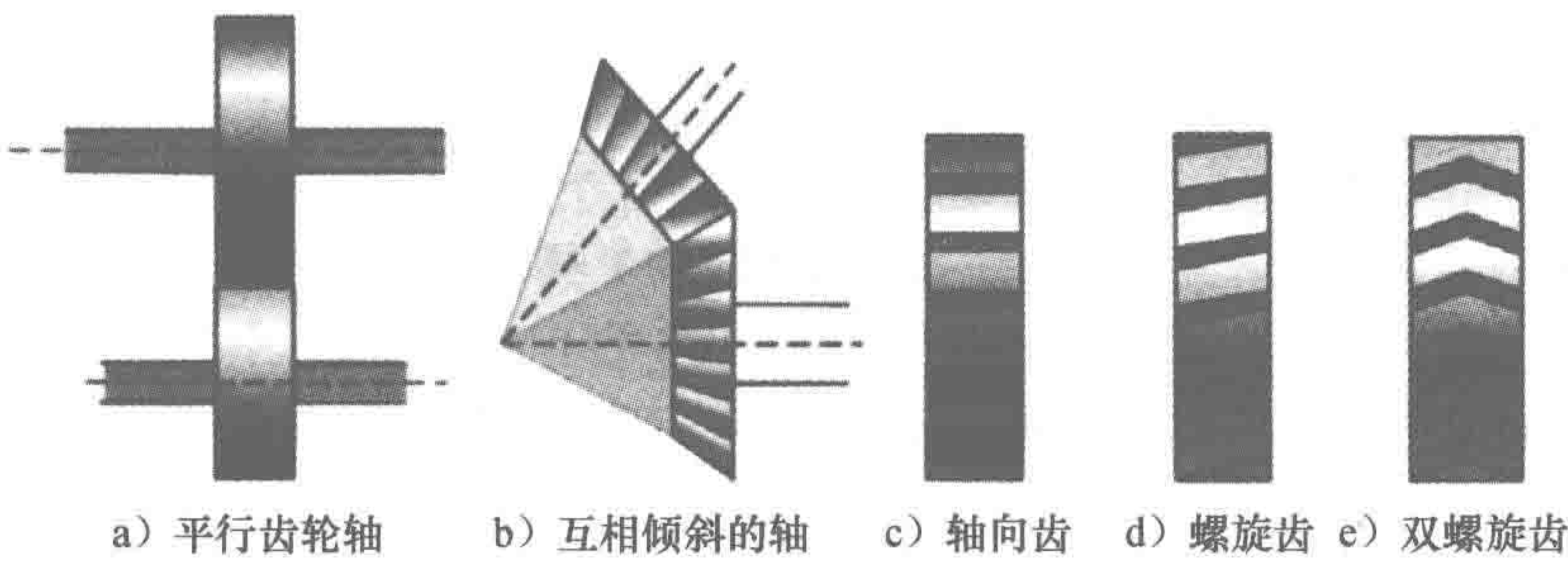


图 8-15　各种齿轮

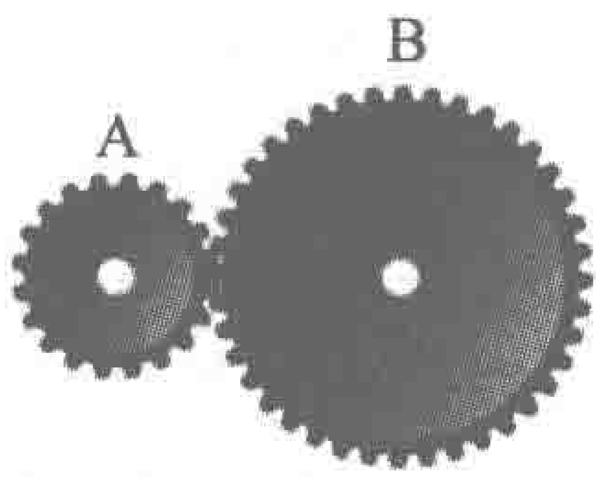


图 8-16　两个啮合的齿轮

由于齿轮的齿数和直径成比例，所以有：

$$\frac{\omega_A}{\omega_B}=\frac{\text{B 齿数}}{\text{A 齿数}}=\frac{d_B}{d_A}$$

因此根据上述数据，B 的直径是 A 的两倍。**齿轮传动比**可用于表示相互啮合的齿轮间的转

速比。上面例子中的齿轮传动比是2。

8.5.1 齿轮系

齿轮系用于描述一系列相互啮合的齿轮。**简单齿轮系**表示这个系统中每根轴只有一个齿轮，如图8-17所示。对于这样的齿轮系，总的齿轮传动比就是输入轴和输出轴的角速度比，即

$$G=\frac{\omega_A}{\omega_C}$$

197

考虑图8-17中的简单齿轮系，A有9齿，C有27齿。因为齿轮的角速度与齿轮的齿数成反比，所以齿轮传动比是27/9 = 3。相比于仅有A、C两个轮的情况，B的作用仅仅是改变C的转动方向。中间的这个轮B称为**惰轮**。

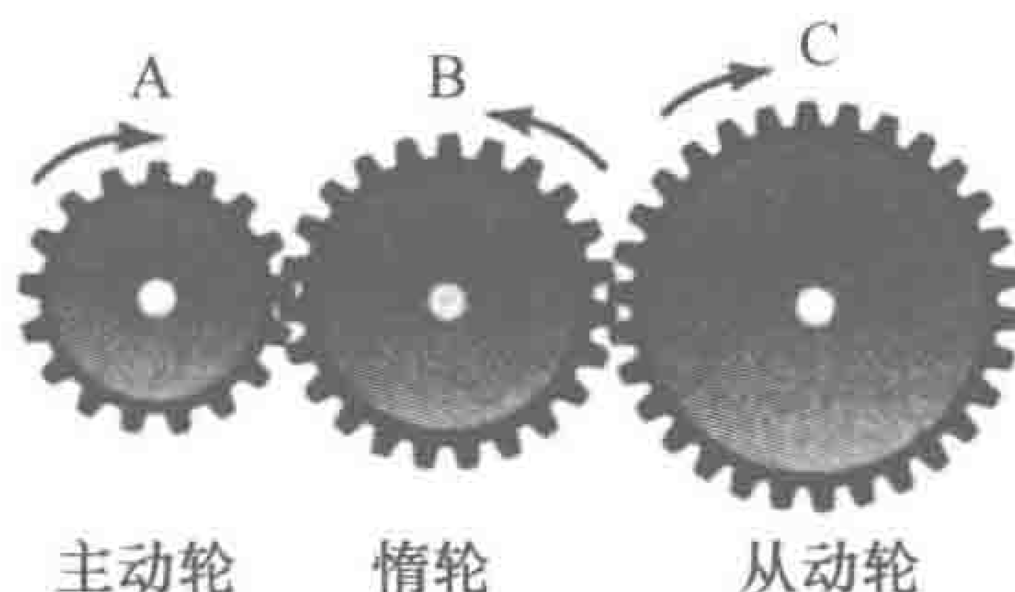


图8-17 简单齿轮系

我们可以重新写出总的齿轮传动比：

$$G=\frac{\omega_A}{\omega_C}=\frac{\omega_A}{\omega_B}\times\frac{\omega_B}{\omega_C}$$

这里$\frac{\omega_A}{\omega_B}$是第一对齿轮间的齿轮传动比，$\frac{\omega_B}{\omega_C}$是第二对齿轮间的齿轮传动比，因此总的齿轮传动比就是相连的各对齿轮传动比的乘积。

复合齿轮系是在一个机构中有两个齿轮安装在一根轴上。图8-18a、b是两个复合齿轮系的例子。图8-18b中的齿轮系使输入和输出轴成一条直线。

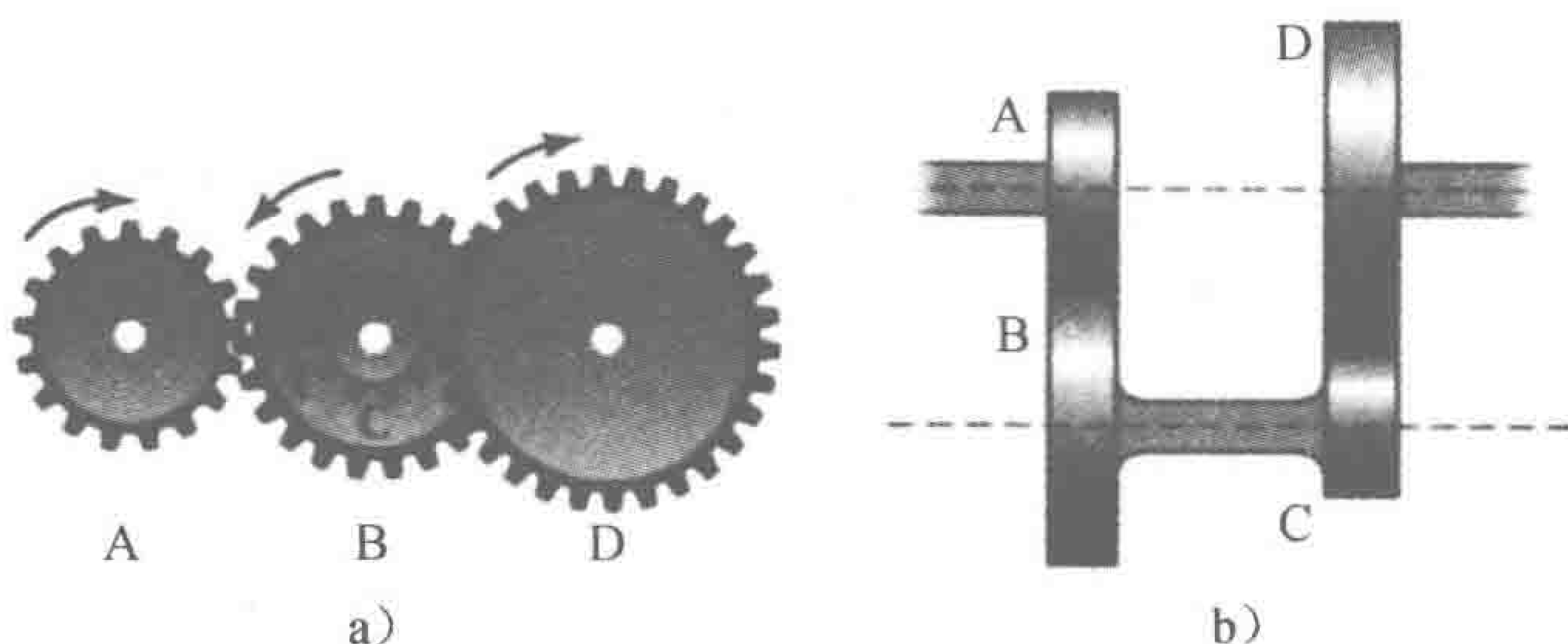


图8-18 复合齿轮系

当两个齿轮安装在同一轴上时就有相同的角速度。因此在图8-18中就有$\omega_B=\omega_C$，总的齿轮传动比是

$$G=\frac{\omega_A}{\omega_D}=\frac{\omega_A}{\omega_B}\times\frac{\omega_B}{\omega_C}\times\frac{\omega_C}{\omega_D}=\frac{\omega_A}{\omega_B}\times\frac{\omega_C}{\omega_D}$$

在图8-18b中，为了保证输出和输入轴共线，还要保证齿轮的半径：

$$r_A+r_B=r_C+r_D$$

考虑图8-18a中复合齿轮系的情况。A有15齿，B有30齿，C有18齿，D有36齿。根据前面的分析可以得到总的齿轮传动比为

$$G=\frac{30}{15}\times\frac{36}{18}=4$$

因此，如果输入A的角速度是160rev/min，那么输出D的角速度是160/4 = 40rev/min。

简单齿轮系无论是直齿轮、斜齿轮，还是锥齿轮，总的齿轮传动比一般都限制在10以内。

198

这是因为齿轮的体积要限制在一定的规模内，而且最小的齿轮还要至少保证一定的齿数（10～20之间）。更大的齿轮传动比可以通过复合齿轮系来实现。因为总的齿轮传动比是通过各对平

行轴齿轮传动比的乘积得到的。

8.5.2　转动和平动的转换

齿轮与齿条（见图 8-19），本质上是另一种齿轮形式，其中一个齿轮具有无限大的半径。这样的齿轮可以完成转动和平动之间的相互转换。

另一种可以完成此功能的机构是**螺杆和螺母系统**。传统的螺杆和螺母的形式一般是螺母转动，同时沿着螺杆移动。然而如果螺杆转动，那么螺母就沿着螺杆的螺纹移动。这种螺杆称为**丝杆**。导程 L 是螺母在螺杆上转动一周时沿着螺杆轴向移动的距离。对于单螺纹螺杆，导程就是螺距。在 n 圈内沿着螺杆轴线移动的距离就是 nL，如果这 n 圈是在时间 t 内完成，那么平行于丝杆轴的线速度就是$\frac{nL}{t}$。由于$\frac{n}{t}$是螺杆每秒转的圈数 f，所以：

$$v = \frac{nL}{t} = fL$$

然而用这种方式来把旋转运动转换为直线运动还是存在一些问题。在螺母和丝杆的直接滑动接触中，存在着很大的摩擦力，而且整体的刚性很差。可以用**滚珠丝杆**来解决摩擦力的问题。原则上这种螺杆和前面的普通螺杆是一样的，但是螺母的螺纹上却安装了球轴承。这种组合已经用于机器人的手臂中（见图 8-20），由一台直流减速电动机驱动滚珠丝杆，使螺母上下移动，然后螺母的移动通过连杆传递到手臂上。

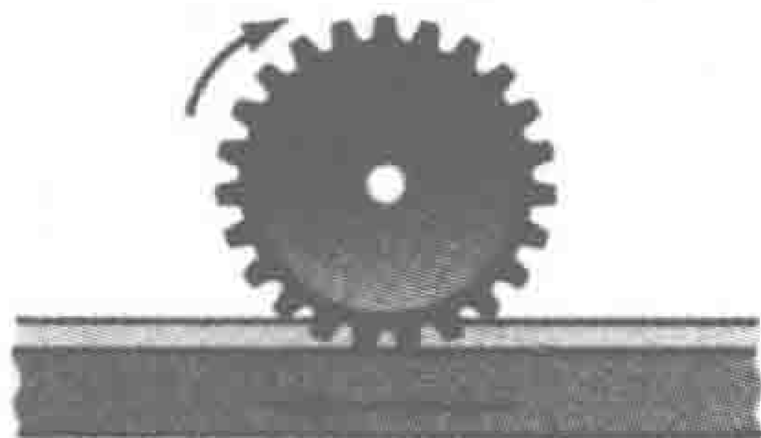


图 8-19　齿轮与齿条

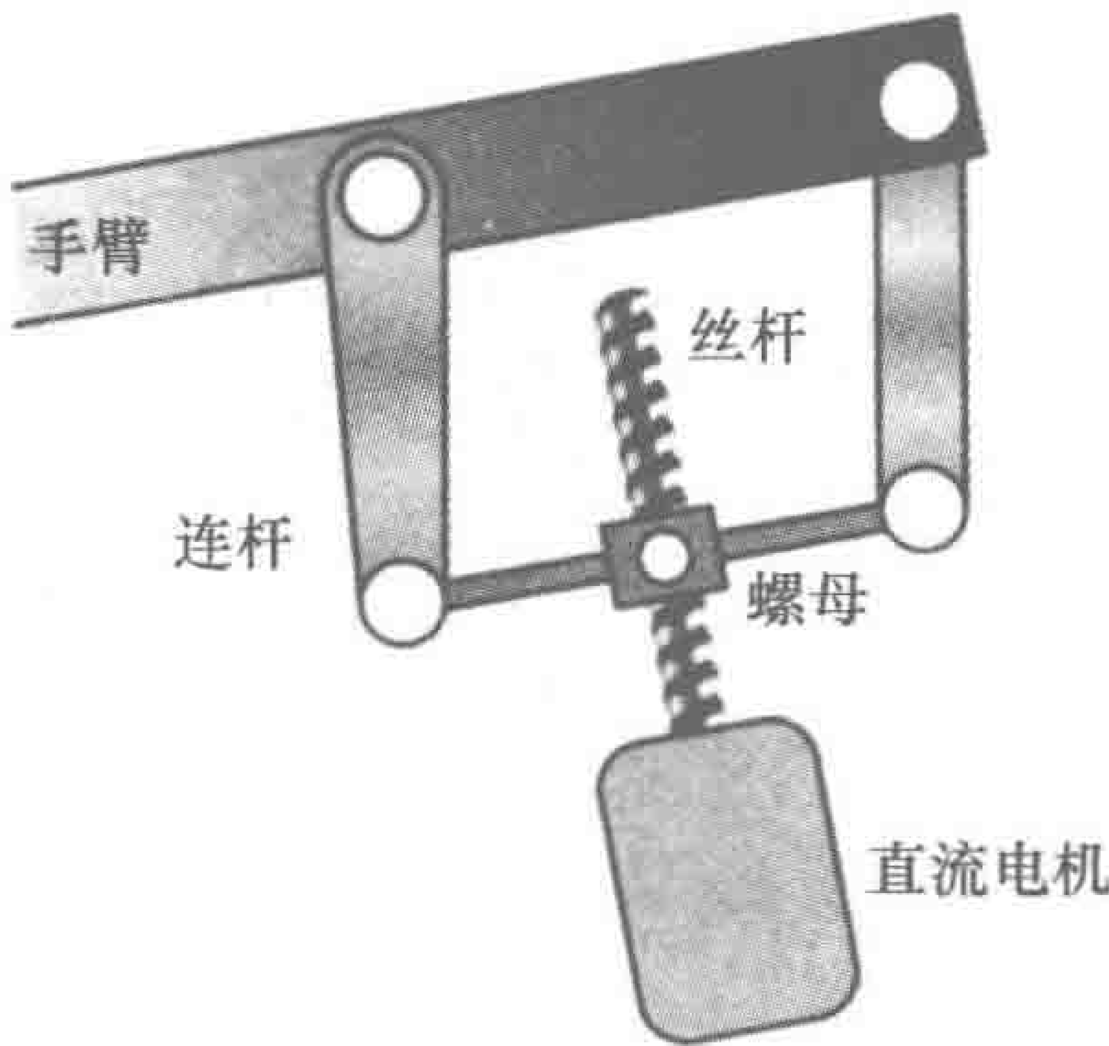


图 8-20　用于移动机器人手臂的滚珠丝杆和连杆

199

8.6　棘轮和棘爪

棘轮可以用于锁住一个带载的机构。图 8-21 就是一个棘轮和棘爪机构。整个机构包括一个**棘轮**，上面有锯齿，可与**棘爪**咬合在一起。棘爪可以前后摆动，用于和棘轮啮合。棘轮的齿形设计成只能令棘轮按照一个方向转动。由于棘爪的限制，图中的棘轮不能顺时针转动，除非把棘爪抬起来。棘爪通常受到一定的弹簧力，以保证它可以一直咬合棘轮。

在鼓上缠绕缆绳的绞盘经常使用这种机构，它可以防止松开手柄时缆绳也跟着松开。



图 8-21　棘轮和棘爪

8.7　带传动和链传动

带传动主要由一对带轮组成，一只带轮通过带把运动传递给另一只带轮（见图 8-22）。带传动利用带、轮间产生的摩擦，以及带和轮子上一段圆弧的接触来传递转矩。由于传递是依靠

摩擦力的，所以常常发生滑动。由于带上不同的张紧程度，所以转矩传递造成带一个边紧一个边松。如果紧边的张力为 T_1，松边为 T_2，图 8-22 中的 A 轮是主动轮，那么：

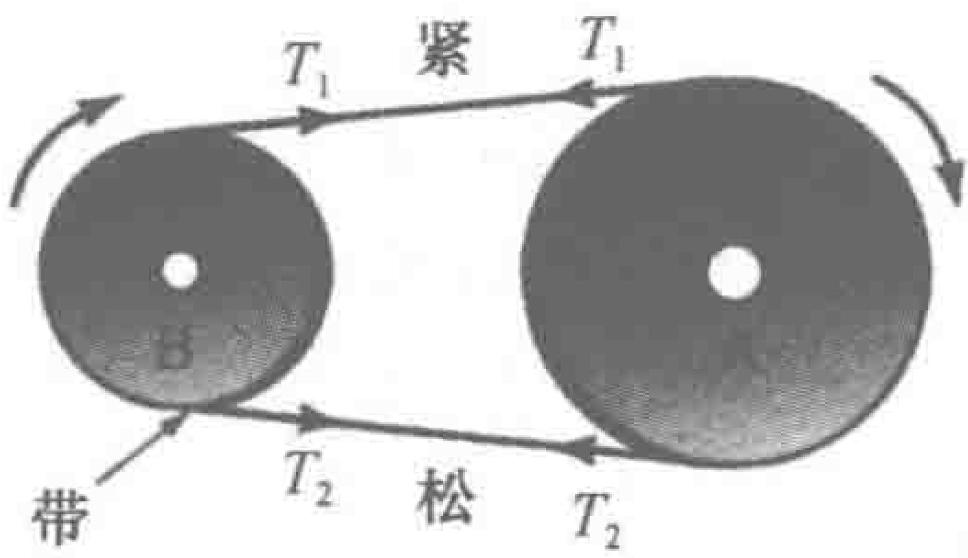


图 8-22　带传动

$$A\text{ 的转矩} = (T_1 - T_2)r_A$$

r_A 是 A 的半径。对于从动轮 B 轮，有：

$$B\text{ 的转矩} = (T_1 - T_2)r_B$$

r_B 是 B 的半径。由于传递的功率是转矩乘以角速度（A、B 角速度分别为 v/r_A 和 v/r_B），v 是带的带速，所以：

$$\text{功率} = (T_1 - T_2)v$$

作为一种在两个转轴之间传递功率的方法，带传动具有一些优点。首先是带的长度可以变化，可以很容易地调整带的长度以适应不同范围内的轴间距离。同时，带传动还可以在过载的时候保护系统，因为一旦负载超出最大限度，摩擦力就无法带动其跟随运动，皮带就会发生打滑，所以就不会损坏整个机构。如果两轴之间的距离比较大，那么使用带传动比齿轮更合适；如果距离比较小，那么使用齿轮更合适。不同大小的带轮可以起到齿轮的作用。然而齿轮传动比一般限制在 3 以内，因为带和带轮之间需要有足够的弧长保持接触。

200

图 8-22 中的带传动说明从动轮和主动轮的转向一致。图 8-23 中给了两种反向带传动的类型。这两种方式中都不能使用 V 形带或者同步带，因为这两种形式下带子的两边都和带轮发生接触。

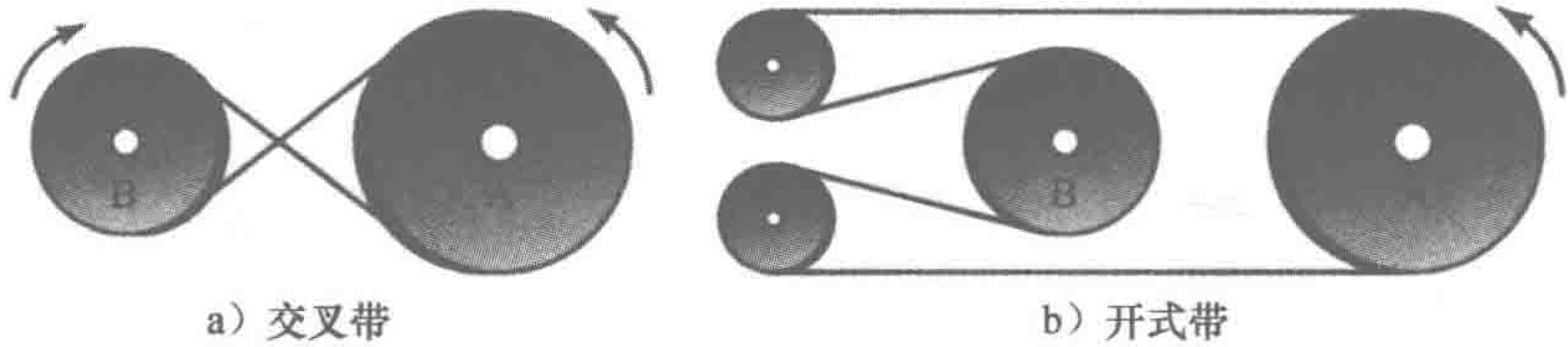


图 8-23　反向带传动

8.7.1　带的类型

带的四种主要类型（见图 8-24）概述如下。



图 8-24　带的类型

1. 平带

平带的横截面是矩形的，它的传送效率可以达到 98%，而且几乎没有噪声。它们可以在两个带轮的轮心之间通过长距离传递功率。边缘突起的带轮可以防止带从轮上脱离。

2. 圆形带

圆形带具有圆形的横截面，通常和槽轮一起使用。

3. V 形带

V 形带和槽轮一起使用，效率比平带低，但是它们中的很多都可以用于单个轮子上，给出多个驱动。

4. 同步带

同步带需要和齿轮一起使用，同步带本身也有齿，它的齿可以和带轮上的齿相互啮合。同

步带和其他类型的带不一样，它不会发生拉伸或打滑，因此它可以以一个恒定的角速度率传递功率。这些齿使其可以快慢不同的速度运行。 201

8.7.2 链传动

使用链传动可以防止打滑，链可以锁入滚动轴的齿上，相当于一对啮合在一起的齿轮。链传动和简单齿轮系的齿轮传动比的关系是一致的。链传动可以使用一个单轮来带动很多转轴，所以可为多个轴提供驱动。它们并不像同步带那样安静地工作，但是却可以带来更大的转矩。

8.8 轴承

只要两个接触面之间出现了相对移动，无论是滚动还是滑动，都会因为摩擦生热而带来能量损失和磨损。**轴承**的作用就是使这两者之间尽量减少摩擦，同时保证相对运动的最大精度。

尤其重要的是，我们需要给转轴以合适的支撑，即支撑径向负载。当相对运动是转动的时候，**推力轴承**用来承受沿着轴向的力。下面将介绍常用轴承的特性。

8.8.1 径向滑动轴承

径向轴承用于支撑径向承载的转轴，术语**轴颈**是指转轴上安装轴承的地方。轴承包括一些合适材料制成的内嵌物，它们位于转轴和支架之间（见图 8-25）。轴的转动使它的表面滑过轴承表面。内嵌物可能是一种白合金、铝合金、铜合金、青铜或者尼龙和聚四氟乙烯（polytetrafluoroethylene，PTFE）这样的聚合物。相比于转轴直接在孔内转动，这种内嵌物可以使相对运动的表面之间的摩擦更小，磨损更少。轴承可以是干摩擦或者带润滑的轴承。尼龙或者聚四氟乙烯这类聚合物一般可不使用润滑，因为这些材料的摩擦系数特别低。烧结青铜是一种广泛应用于轴承的材料，因为它有多孔的结构，可以使润滑油浸入到轴承中，从而形成一种“内置”润滑剂。

可能的润滑形式有以下几种。

1. 液体动压润滑

液体动压滑动轴承中的转轴持续在油液中转动，不直接和金属接触（见图 8-26）。转轴转动在油液中产生压力来支撑负载。

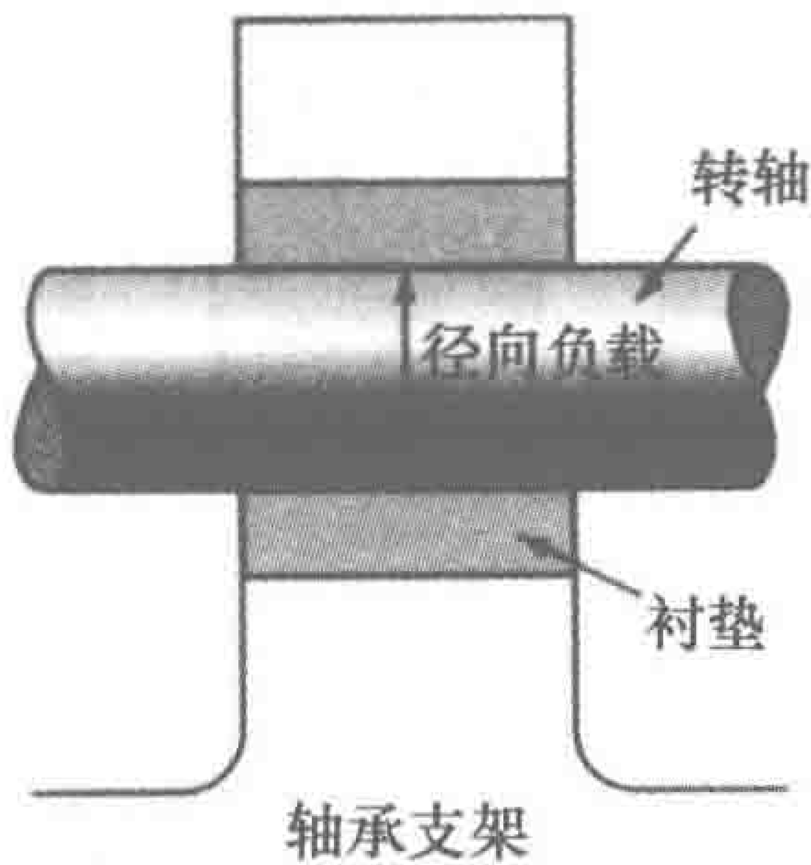


图 8-25 径向滑动轴承

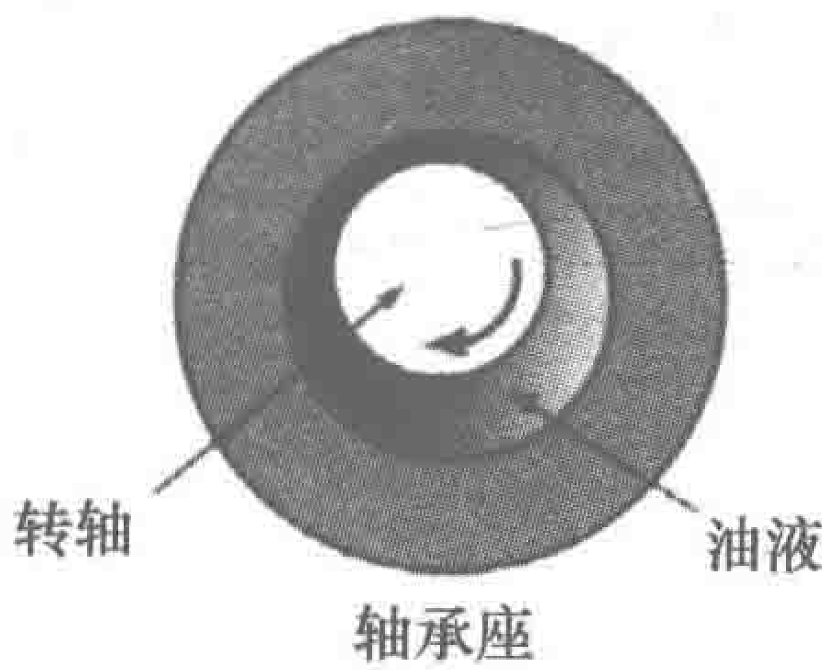


图 8-26 液体动压滑动轴承

2. 液体静压润滑

液体动压润滑的问题之一是转轴仅仅在转动的时候在油液中，其余时候金属之间的表面直接接触。为了避免在低负载和刚启动时产生过大的摩擦力，润滑油被足够大的压力压入轴承的承载部分，这样就可以在静止的时候把转轴抬起来，与支架脱离接触。 202

3. 固体薄膜润滑

该方法使用固体材料涂层，比如石墨或者二硫化钼。

4. 边界层润滑

这是一种附着在轴承表面上的润滑薄层。

8.8.2 球轴承和滚柱轴承

使用这种轴承可使从转轴到支架的负载传递通过滚动接触而不是滑动接触。滚动轴承包括四个主要部分：内圈、外圈、滚动体（球或滚柱）和保持架（见图 8-27）。内圈、外圈有坚硬的轨道，滚动体在里面滚动。

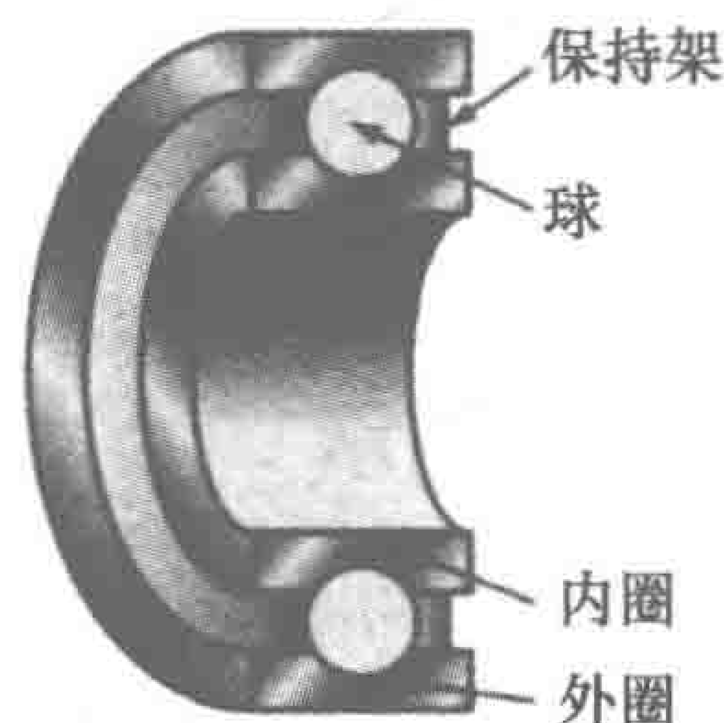


图 8-27 球轴承的组成

球轴承有以下几种类型。

1. 深沟球轴承

这种轴承擅长支撑径向负载，但是对轴向负载能力有限。这是一种通用性很强的轴承，可以在很宽的负载和转速范围内使用（见图 8-28a）。

2. 装填槽球轴承

这种轴承可以比深沟球轴承承受更大的径向负载，但是不能用在轴向负载的场合中（见图 8-28b）。

3. 角接触球轴承

这种轴承可以同时承受轴向和径向负载，而且承受轴向负载的能力强于深沟球轴承（见图 8-28c）。

4. 双列球轴承

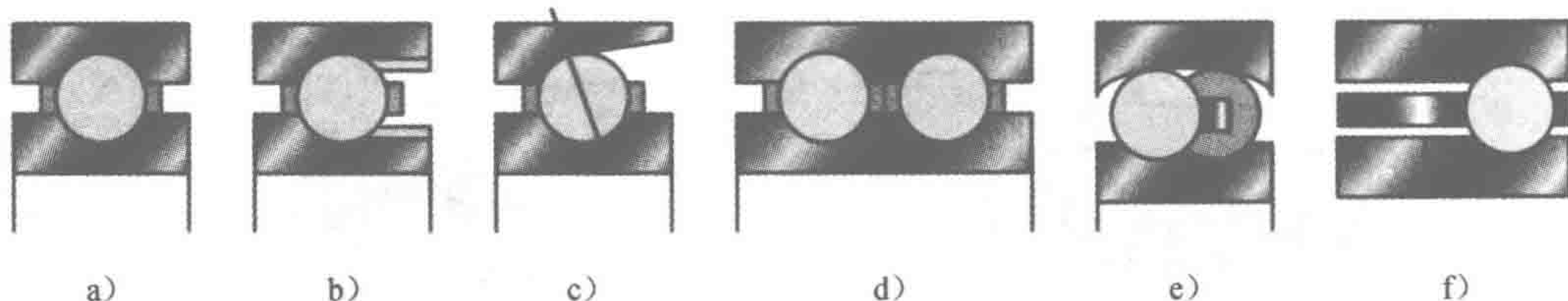
这种轴承有很多形式，相比于它们的单列形式，双列球轴承可以承受更大的径向负载。图 8-24d 中所示的是双列深沟球轴承，上述每种轴承都有它们对应的双列形式。

5. 自准球轴承

单列轴承可以承受少量的轴装配误差，但是对可能产生严重轴装配误差的场合可使用自准球轴承。这种轴承只能承受有限的径向负载，而且对轴向负载的承受能力相当差（见图 8-28e）。

6. 推力沟道球轴承

这种轴承用于承受轴向负载，但是并不适合承受径向负载（见图 8-28f）。



203

图 8-28 球轴承的类型

此外还有很多种滚柱轴承，比较常见的有以下几种。

1. 直滚柱轴承

相比于同一尺寸的球轴承，它们在承受径向负载方面更出色，但一般不适合于轴向负载（见图 8-29a）。因为它们的滚动体和滑道之间的接触面更大，所以它们可以比相同大小的球轴承承受更大的负载，但是它们对错位并不具备容错能力。

2. 锥形滚柱轴承

它们的径向负载能力很强，同时单方向的轴向负载能力也不错（见图 8-29b）。

3. 滚针轴承

这种轴承的滚动体是长度/直径比很大的滚针，如果没有足够的空间使用球轴承和滚柱轴

承，就可以使用这种轴承（见图 8-29c）。

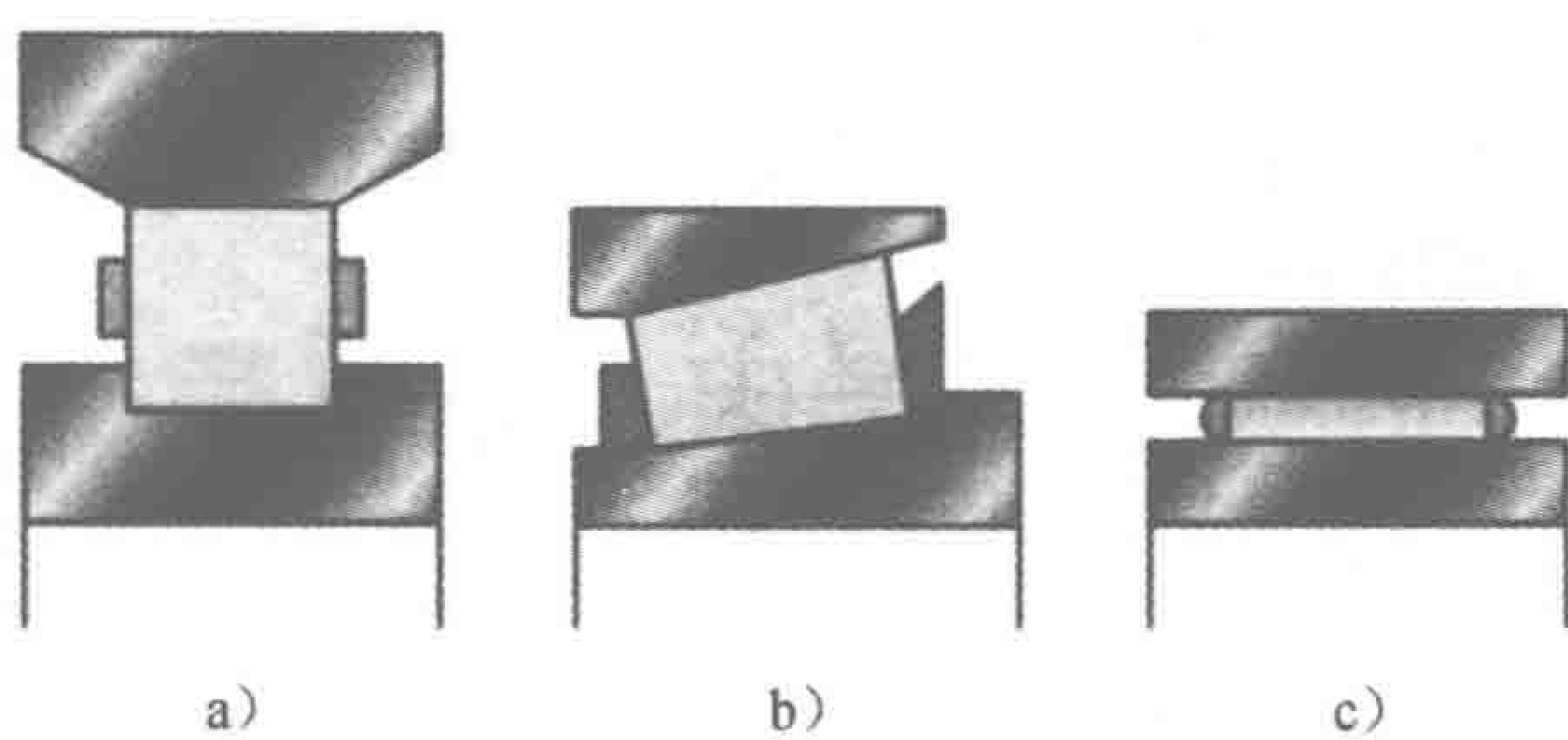


图 8-29　滚柱轴承

8.8.3　轴承的选择

一般地，干滑动轴承仅仅用于转轴直径很小、转速低、负载小的场合；球轴承和滚柱轴承可用于转速更高、负载更大的场合，而且转轴的直径范围也可更宽；而液体动压轴承则常用于转轴直径很大的高负载情况。

本章小结

机构是运动的转换器，可把运动从一种形式转换成其他需要的形式。

物体的运动可以考虑成平动和转动的结合。**自由度**数是产生确定运动时所必需的独立运动分量的数目。

机构中和其他零件在运动上发生相对关系的每个零件，都称为**连杆**。连杆与其相邻连杆连接的点称为**节点**。两个或更多连杆在它们相连处的节点称为**关节**。一系列的关节和连杆称为**运动链**。四杆链由四根连杆组成，它包含四个关节，在关节处可以发生转动。

凸轮是可以转动或摆动的物体，**从动件**与凸轮相接触，可以随着凸轮而往复运动或摆动。 204

齿轮可以在平行轴和相互成角度的轴之间传递转动。

齿条与齿轮和**螺杆与螺母**系统可以把转动转换成平动。

当机构正在支撑负载时，**棘轮**可以锁住它。

带传动和**链传动**可以在有一定距离的平行轴之间传递转动。

轴承可以使一个零件相对于另一个零件运动时尽量减少摩擦，并且同时保证相对运动的精度。

习题

8.1　解释名词：(a) 机构，(b) 运动链。

8.2　解释什么是四杆链。

8.3　考察下面的机构，并说出每一种机构的自由度数目。

(a) 汽车发动机罩铰链
(b) 旅行车的后挡板
(c) 汽车挡风玻璃的雨刷
(c) 你的膝盖
(e) 你的脚踝

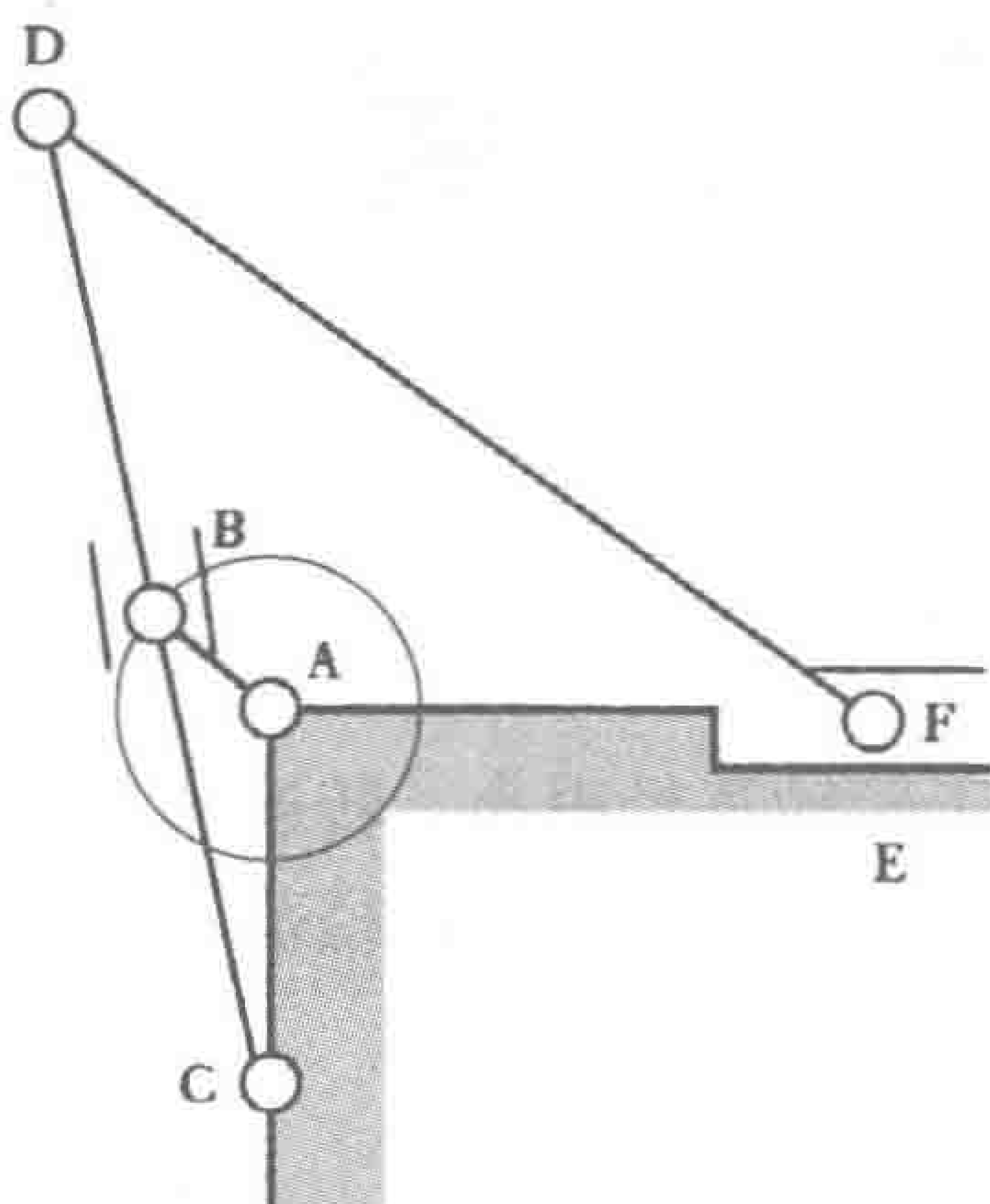
8.4　分析下面机构的运动，并且说出它们是纯转动、纯平动，还是转动和平动的结合。

(a) 计算机键盘的按键
(b) XY 平面绘图机上的笔
(c) 钟表的时针

(d) 动圈式安培计的指针

(e) 自动螺丝起子

8.5 图 8-30 中所示机构，臂杆 AB 以恒定转速旋转，B 和 F 是分别沿着 CD 和 AF 滑动的滑块。描述这个机构的运动。



205

图 8-30　习题 8.5

8.6 描述图 8-31 中的从动件位移如何随着凸轮转角的变化而变化。

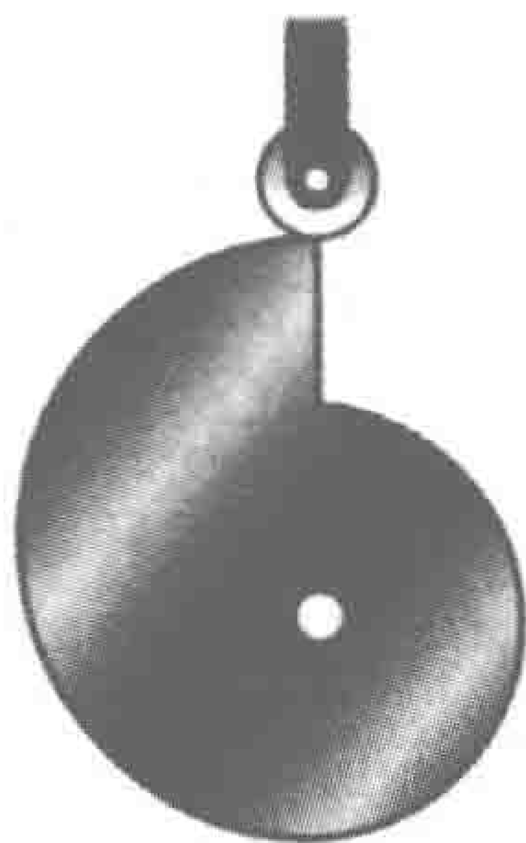


图 8-31　习题 8.6

8.7 一个直径为 100mm 的圆形偏心凸轮，其旋转中心距离圆心 30mm。一个刀形从动件的移动路线穿过凸轮旋转中心，求从动件最大位移和最小位移之差。

8.8 设计一个凸轮从动件系统，使从动件在 40 ~ 100mm 的位移范围内保持速度恒定。

8.9 设计可以满足以下用途的机构：

(a) 按一定时序操作一系列微动开关。

(b) 在一个方向上以恒定速度移动一个工具，然后快速使其回到初始点。

(c) 把转动转换为简谐运动的直线往复运动。

(d) 把在某个角度内的转动转化成线性位移。

(e) 把一根轴的转动传递给另一根轴，两根平行轴之间有一定的距离。

(f) 把一根轴的转动传递给另一根轴，两根轴之间距离很近，且成直角。

8.10 有一个复合齿轮系，最终从动齿轮有 15 个齿，与之啮合的倒数第二级齿轮 90 齿，同一根轴上还有一个 15 齿的齿轮，这个齿轮与有 60 齿的第四个齿轮，即第一主动齿轮相啮合，那么总的齿轮传动比是多少？

206

第9章　电气驱动系统

本章目的

通过本章的学习，读者将能够：

- 评估以下几种电气驱动系统的操作特点：继电器、固态开关（晶闸管、双极性晶体管和MOSFET）、螺线管、直流电动机、交流电动机和步进电动机。
- 解释直流电动机的原理，包括直流永磁电动机和如何控制它的速度。
- 解释无刷永磁直流电动机的原理。
- 解释可变磁阻、永磁体、混合步进电动机的原理和步进控制的顺序是如何产生的。
- 解释电动机选择的惯量匹配、转矩和功率要求。

9.1　电气系统

在电气系统被用做驱动器来实现控制的任何讨论中，包括：

1）开关器件，例如机械开关（如继电器）、固态开关（如二极管、晶闸管和晶体管），可通过控制信号来开启或关闭一些电气设备，如一台加热器或一台电动机。

2）螺线管型器件，可通过螺线管的电流来驱动一个软铁心，例如一个螺线管操作的液压阀或气动阀，通过螺线管的电流来控制液流或气流。

3）驱动系统，例如直流和交流电动机，通过电流可产生旋转。

本章概述这些设备和它们的特性。

9.2　机械开关

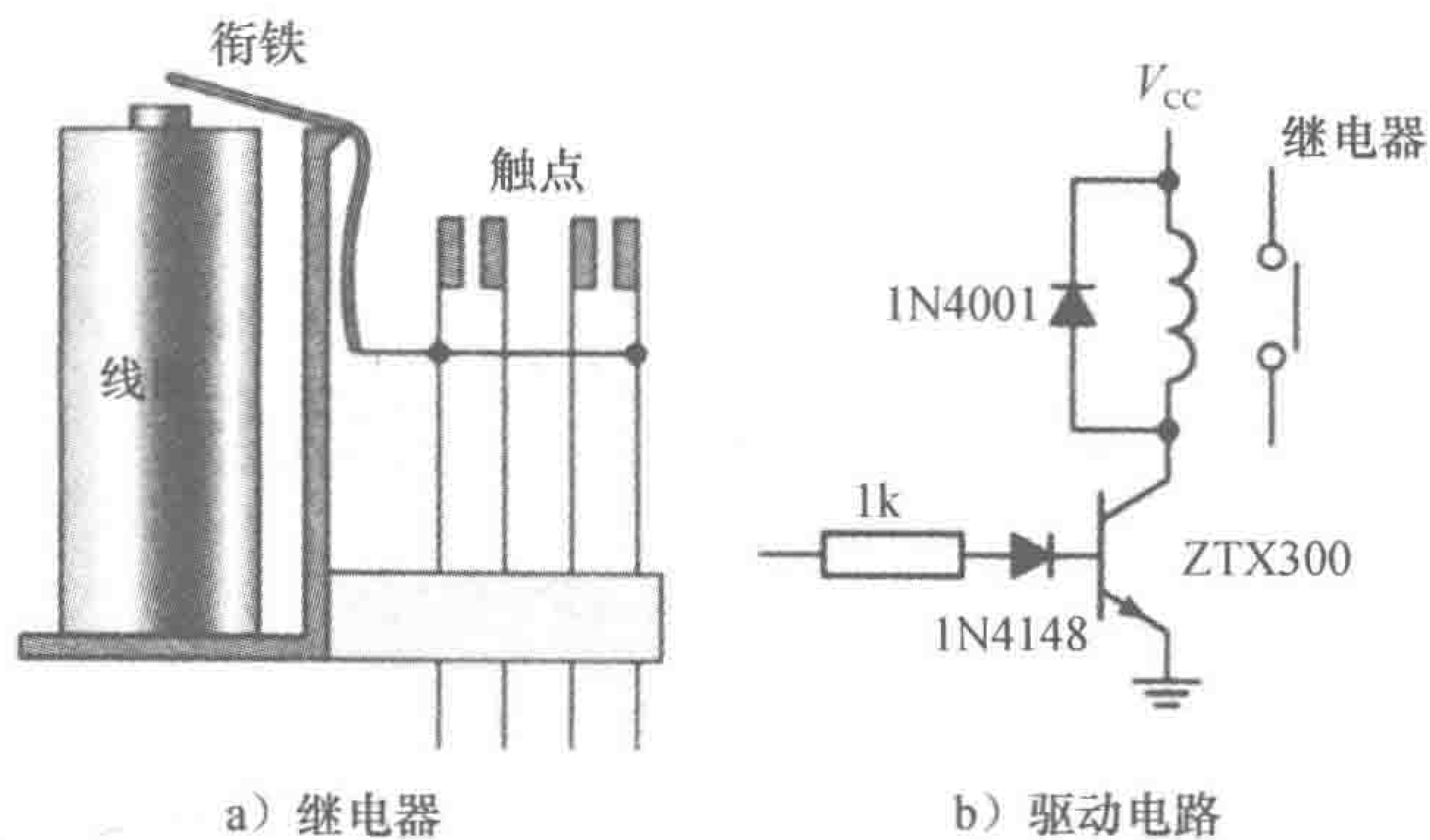
机械开关是一些经常用做传感器来向系统输入信号的元件（见2.12节），例如键盘。本章中，我们关注它们作为驱动器来开启电动机或加热器，或者接通电磁阀的电流来控制液压缸或气缸。**继电器**是一个把机械开关作为驱动器用于控制系统的例子。

9.2.1　继电器

继电器是电控开关，通过它改变一个电路中的电流来开关另外一个电路中的电流。如图9-1a所示的继电器，当有电流通过继电器的线圈时，产生磁场吸引铁心，移动推杆，以关闭常开（NO）触点，打开常闭（NC）触点。

继电器常用于控制系统中。控制器的输出是一个相对较小的电流，所以经常把它和晶体管连接起来，接通继电器线圈的电流，并用这个继电器的触点去接通更大的电流，打开或关闭最后的执行元件，例如温度控制系统中的电加热器或电动机。图9-1b给出了可能用到的电路形式。因为继电器的线圈是感性元件，所以当供电电流被断开或者当晶体管的输入信号由高电平变为低电平时，继电器的线圈能产生一个反向电动势，因此，连接电路中会有损坏发生。为了解决这个问题，继电器线圈两端需要并联一个二极管。当产生反向电动势时，二极管导通，并将线圈短路。这样的二极管称为**续流**（flyback）**二极管**或**飞轮**（free-wheeling）**二极管**。

作为继电器控制系统的例子，图9-2给出了如何用两个继电器来控制气动阀门，使三个气缸A、B、C的活塞依次运动。具体步骤概述如下：



a）继电器 b）驱动电路

图 9-1 继电器原理

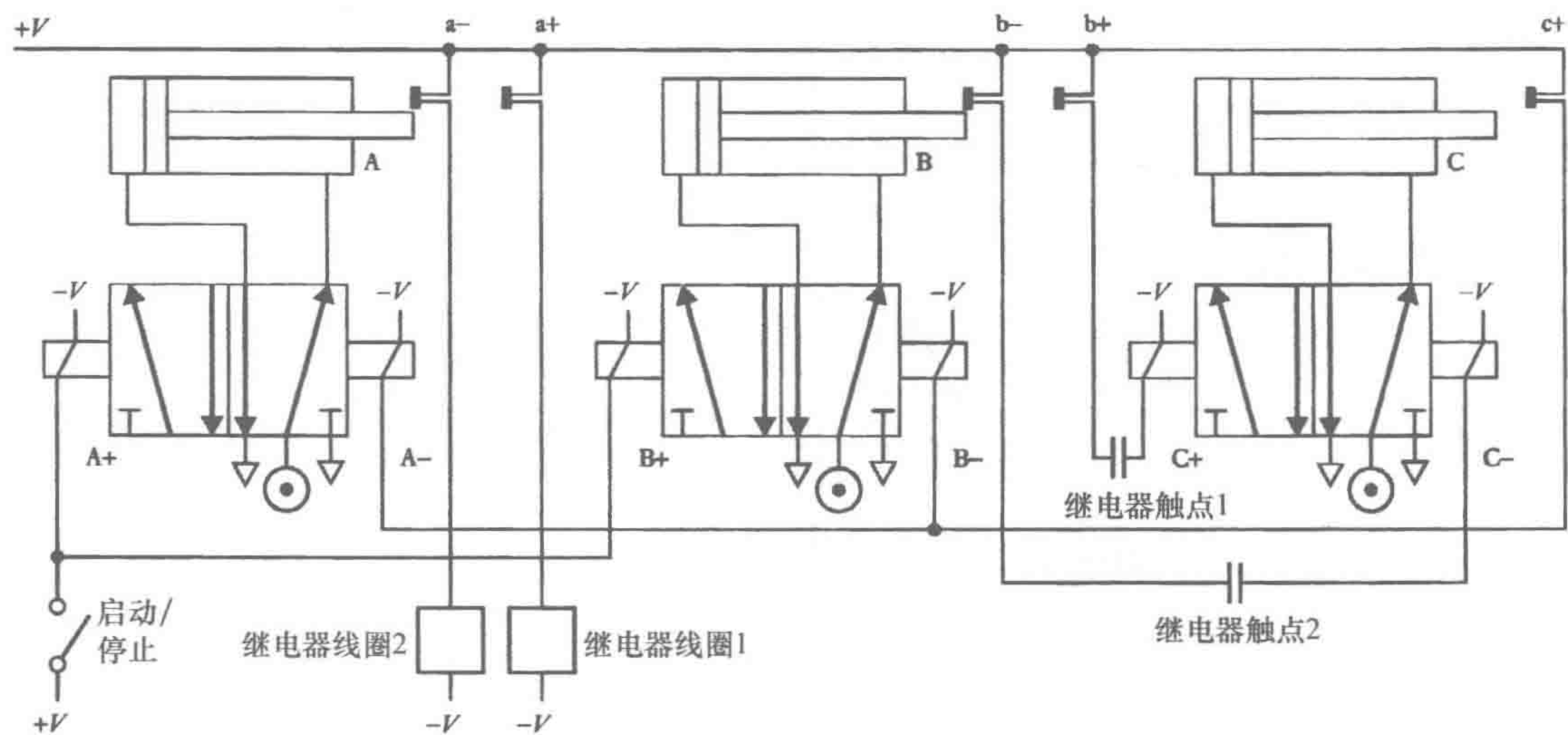


图 9-2 继电器控制系统

1）当启动开关闭合时，电流通过螺线管 A、B，使气缸 A 和 B 伸出，即 A+、B+。

2）然后限位开关 a+和 b+闭合，a+的闭合导致电流流入继电器线圈 1，导致它的触点闭合，因此提供给螺线管 C 电流，并使气缸 C 伸出，即 C+。

3）它的伸出使限位开关 c+闭合，接通 A 和 B 控制阀，使气缸 A 和 B 缩回，即 A−和 B−。

4）限位开关 a-关闭，电流通过继电器线圈 2。它的触点闭合，使电流通过电磁阀 C，并使气缸 C 缩回，即 C−。

208

这个系统的工作顺序是 A+和 B+同时动作，接着 C+动作，然后 A−和 B−同时动作，最后 C−动作。

延时继电器是一个有延迟开关动作的控制继电器。延迟的时间通常是可调节的，可以在电流流入继电器线圈或停止流入线圈时初始化。

9.3 固态开关

很多固态开关可用来切换电路，包括：

1）二极管。

2）晶闸管和三端双向可控硅元件。

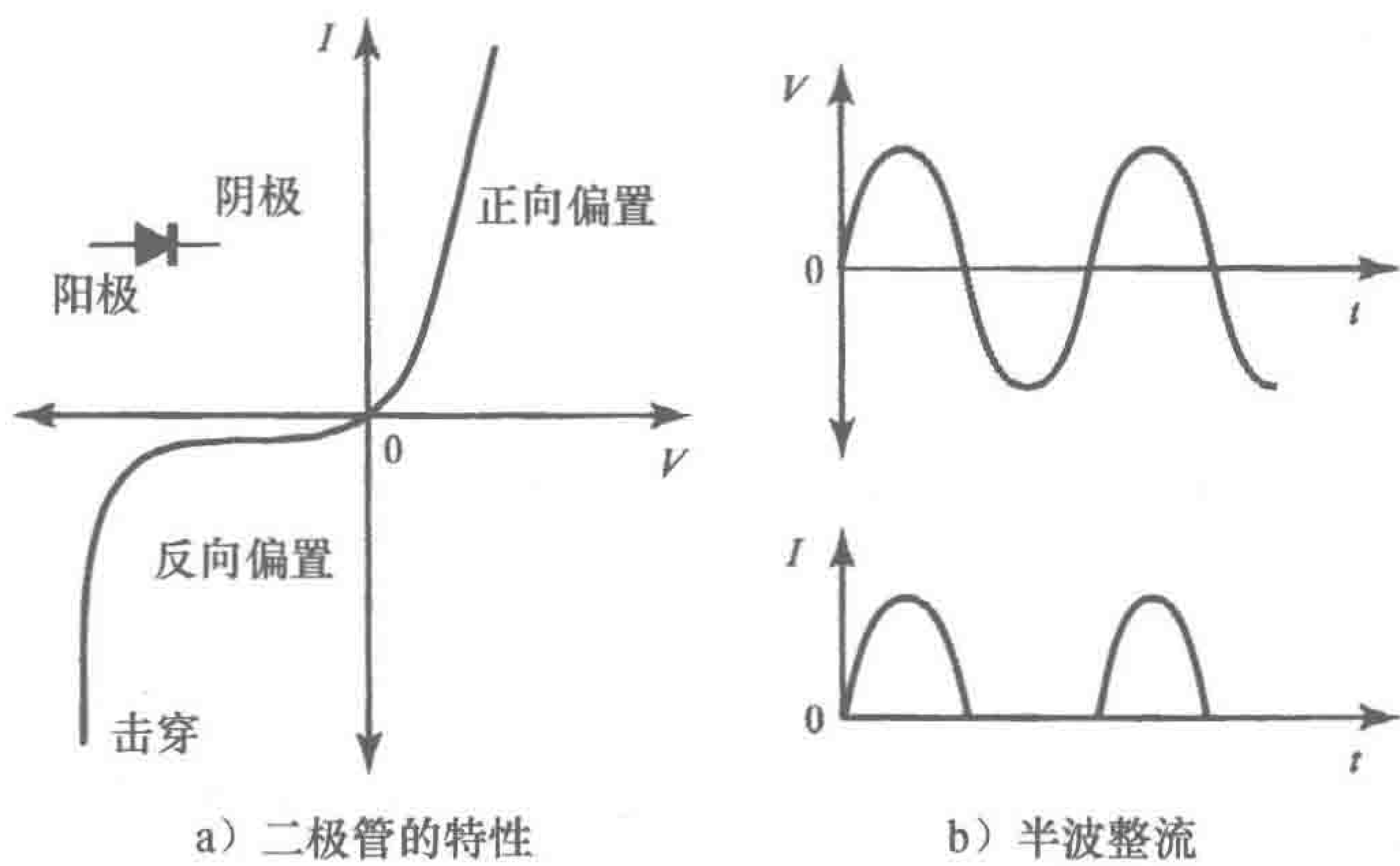
3）双极性晶体管。

4）功率金属氧化物半导体场效应晶体管（MOSFET）。

9.3.1 二极管

二极管伏安特性如图9-3a所示。只有当二极管承受正向偏压时，才通过电流，即阳极相对于阴极为正。如果二极管反向偏压足够大，即一个很高的反向电压，它将会损坏。如果将交流电压加到二极管上，它将在正向偏压时导通，在反向偏压时截止。结果电流通过二极管被半波整流，成为由于输入电压的正半部而产生的电流（见图9-3b），即电路仅在正半个周期导通。

209



a）二极管的特性　　b）半波整流

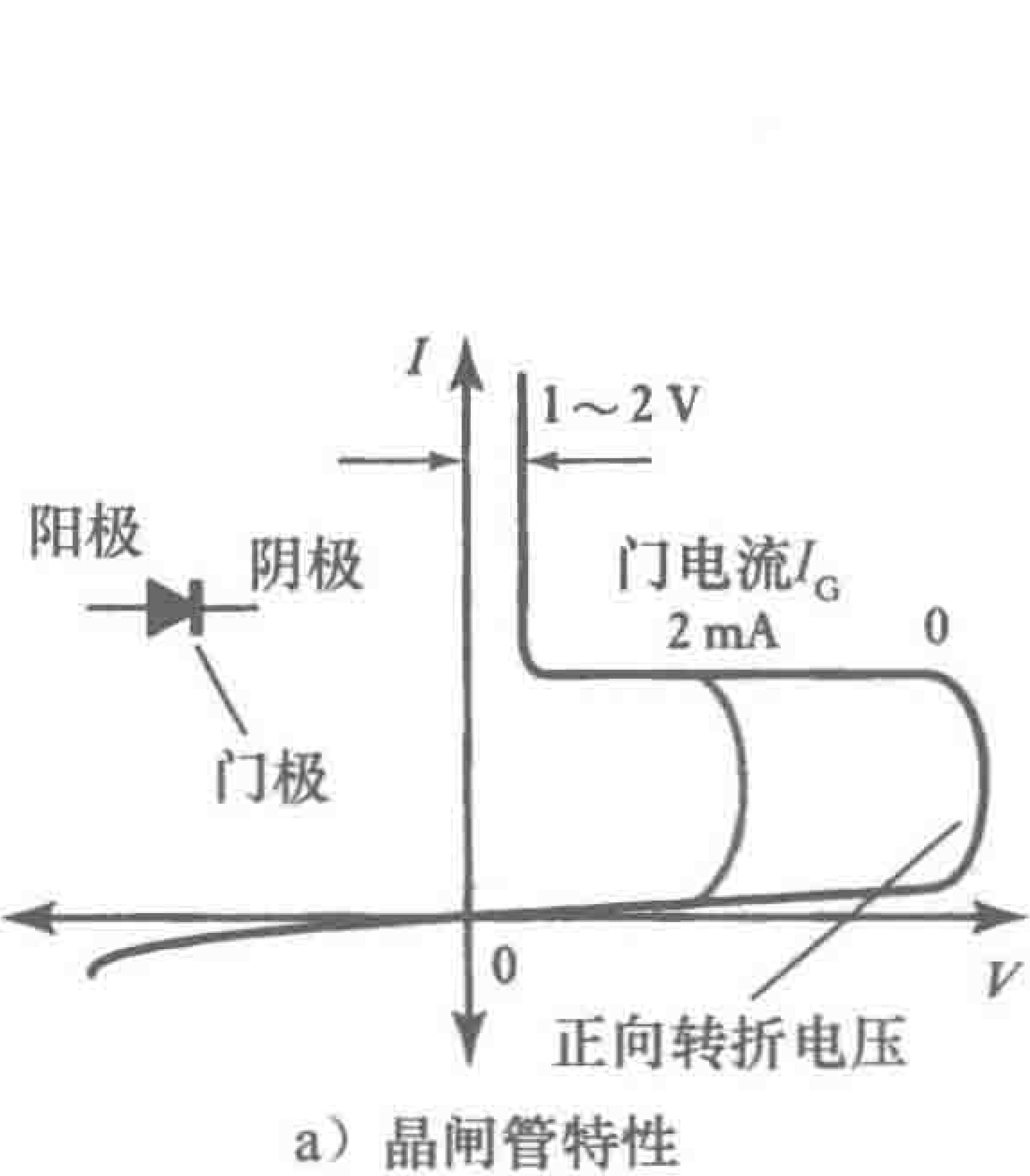
图9-3　二极管

9.3.2 晶闸管和三端双向可控硅元件

晶闸管（thyristor）或**硅控整流器**（SCR）可以看作一个二极管，这个二极管有一个能控制二极管导通的门（控制级）。图9-4a给出了晶闸管的伏安特性曲线。当门电流 I_G 为零时，在阳极阴极间逆向偏压的条件下，晶闸管通过的电流 I 可以忽略（除非有足够大的逆向偏压，如数百伏的电压，使之击穿）。当阳极阴极间正向偏压且控制级阴极间加正向触发电压（门电流 $I_G>0$）时，电流 I 也可以忽略，直到正向电压超过正向转折电压。当这种情况发生时，二极管的电压下降到一个低电平，大约1~2V，此时在电路中电流 I 仅仅被外部电阻所限制。因此，举个例子，如果正向转折电压为300V，当达到了这个数值时，此晶闸管开通并且电压回落到1~2V。如果这个晶闸管和一个假定20Ω的电阻串联（见图9-4b），在击穿之前有一个非常大的电阻（指未发生转折的晶闸管）与此20Ω的电阻串联，所以表面上所有的300V都落在这个晶闸管上，并且电流可忽略。当发生正向转折时，晶闸管两端的电压降落到假定的2V，所以有300 - 2 = 298V电压加在这个20Ω的电阻上，因此此时电流上升到298/20 = 14.9A。晶闸管一旦导通，就保持导通状态，直到正向电流降到几毫安以下。正向转折电压由进入门的电流 I_G 决定：电流 I_G 越大，转折电压越低。晶闸管的功率控制能力很强，因此它广泛应用于切换高功率设备上。例如，德州仪器公司（Texas Instruments）的CF106D有一个最高的断开状态电压400V和一个最高的门触发电流0.2mA。

三端双向可控硅开关元件（triac）和晶闸管相似，等同于在一个芯片上有一对反并联的晶闸管。无论在正向或反向偏压情况下，三端双向可控硅开关元件都能被开启。图9-5给出了它的特性。举个例子，摩托罗拉（Motorola）MAC212-4三端双向可控硅开关元件有一个最高的断开状态电压200V和一个最高的开启状态电流12A有效值（r. m. s）。三端双向可控硅开关元件

是简单的、相对便宜的控制交流电源的方法。



a）晶闸管特性

20 Ω
V
门极

b）晶闸管电路

图 9-4　晶闸管

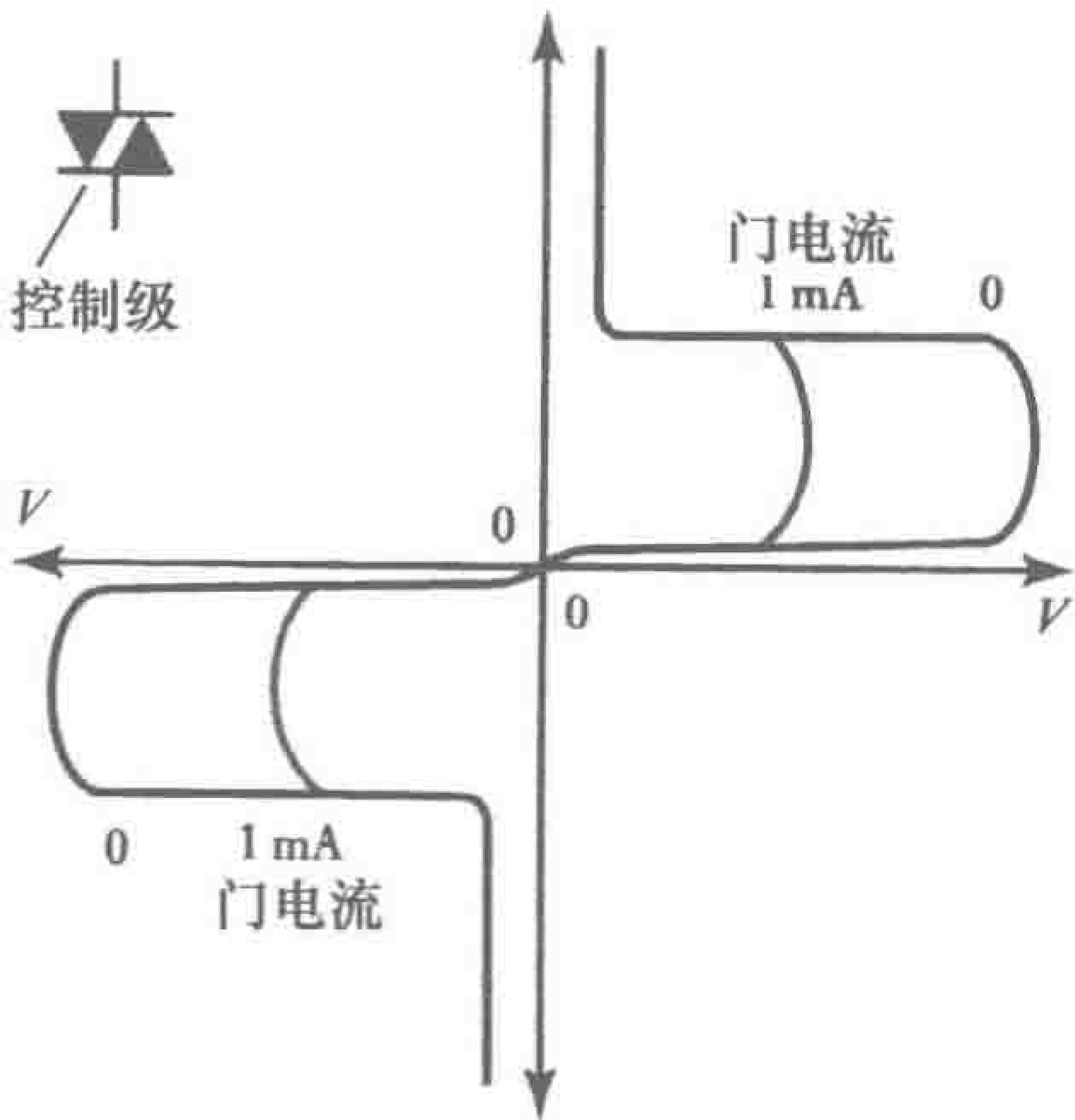
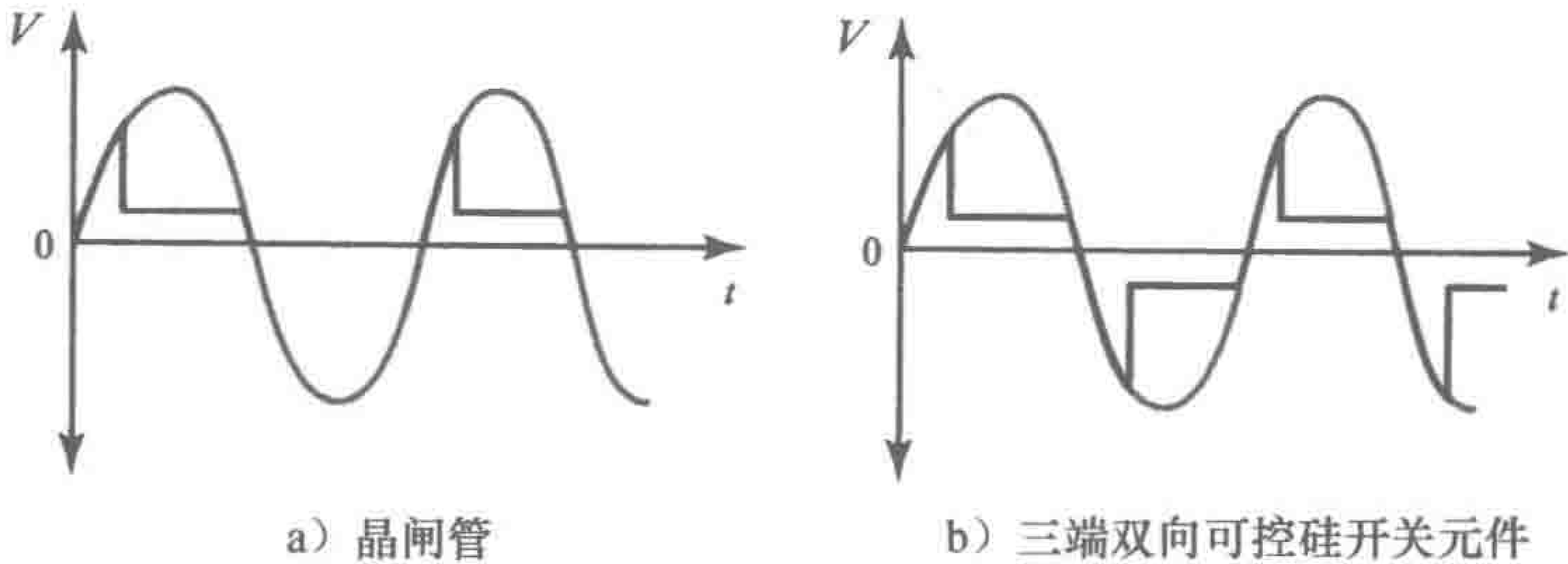


图 9-5　三端双向可控硅开关特性

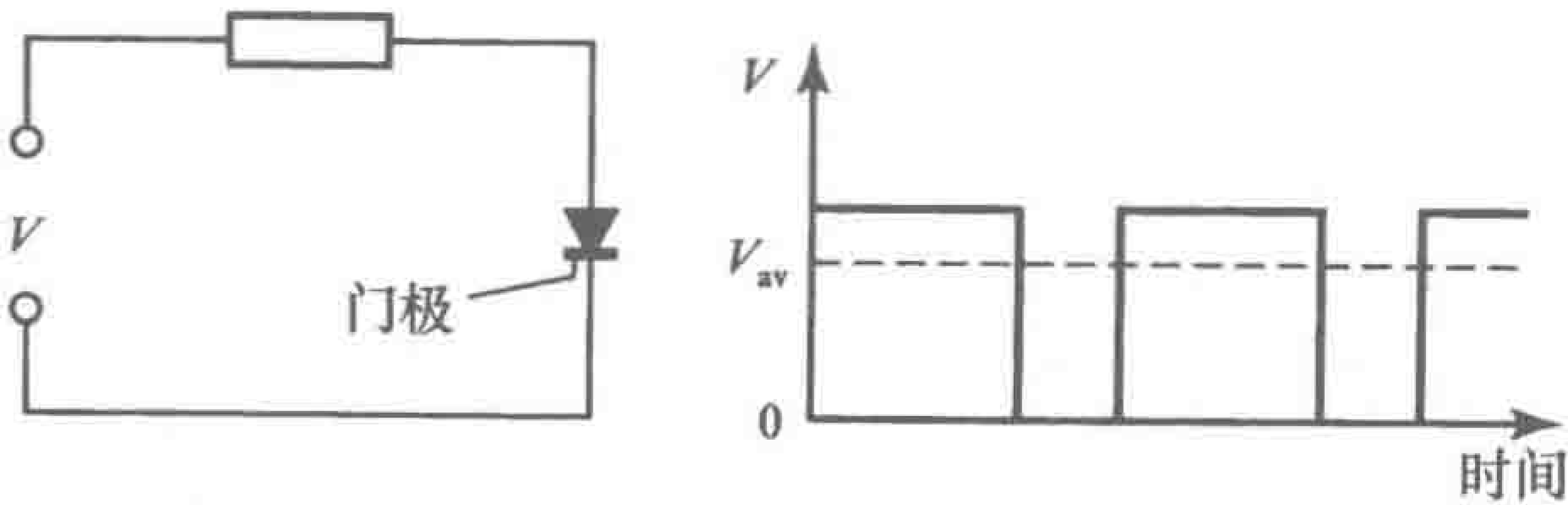
图 9-6 给出了正弦交变电压加在晶闸管和三端双向可控硅开关元件上的效果。当电压达到正向转折值时发生正向导通，之后通过此器件的电压维持在低值。



a）晶闸管　　b）三端双向可控硅开关元件

图 9-6　电压控制

怎样将这样的器件用于控制呢？图 9-7 给出了一个用晶闸管控制直流电压 V 的例子。图中，这个晶闸管被用做一个开关，通过门极控制器件的开关。通过对门极使用交变信号，供电电压会被斩波，并产生一个间断的电压。于是，改变直流输出电压的平均值，并由施加到门极上的交变信号控制。



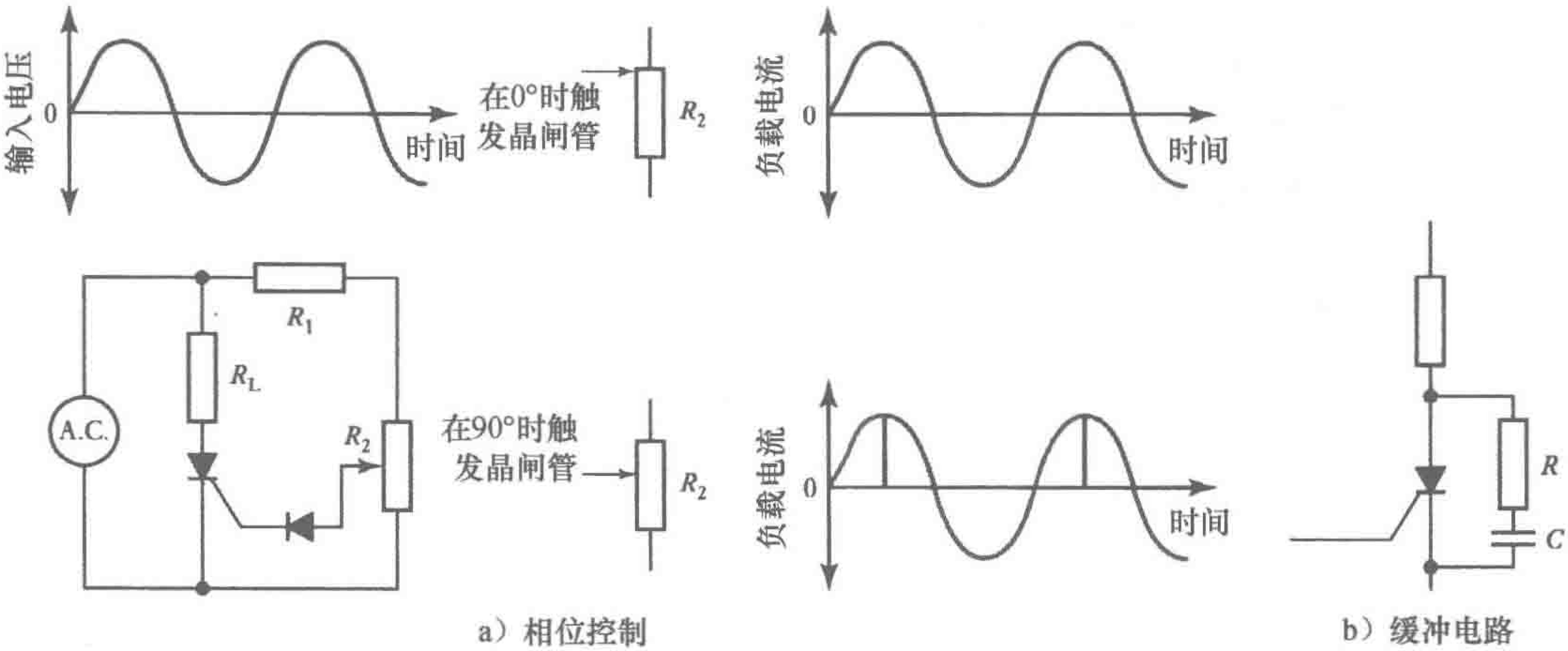
210～211

图 9-7　晶闸管直流控制

另一个控制例子是改变电热器、电动机或灯调光器的电流。图 9-8a 给出了一个半波的变阻相位控制电路。交变电流加载到负载上，例如调光电路中的灯 R_L 和一个晶闸管串联。R_1 是一个限流电阻，R_2 是一个分压器，其设定晶闸管的触发电平。此处的二极管是为了防止交变电压的反向部分而加载到门上。通过调节 R_2，晶闸管可在交变电压正半周期的 0°～90°时触发。当晶闸管在周期开始时（即 0°）触发，它导通整个正半周期，并且将最大的功率加到负载上。随着晶闸管触发的相位向周期后延迟，加载到负载上的功率减少。

当源电压突然加载到晶闸管或三端双向可控硅元件（控制级关闭）上时，晶闸管可能由

关变为开。产生这种现象的典型电压改变速率约为50V/μs。如果电压源是直流电压，则晶闸管能保持在这种导通状态直到电路被断电。为了防止电源电压突变而产生此种现象，电压随时间变化的速率，即 dV/dt，通过一个**缓冲电路**（snubber circuit）来控制。图 9-8b 是电阻与电容串联，再与晶闸管并联的缓冲电路。

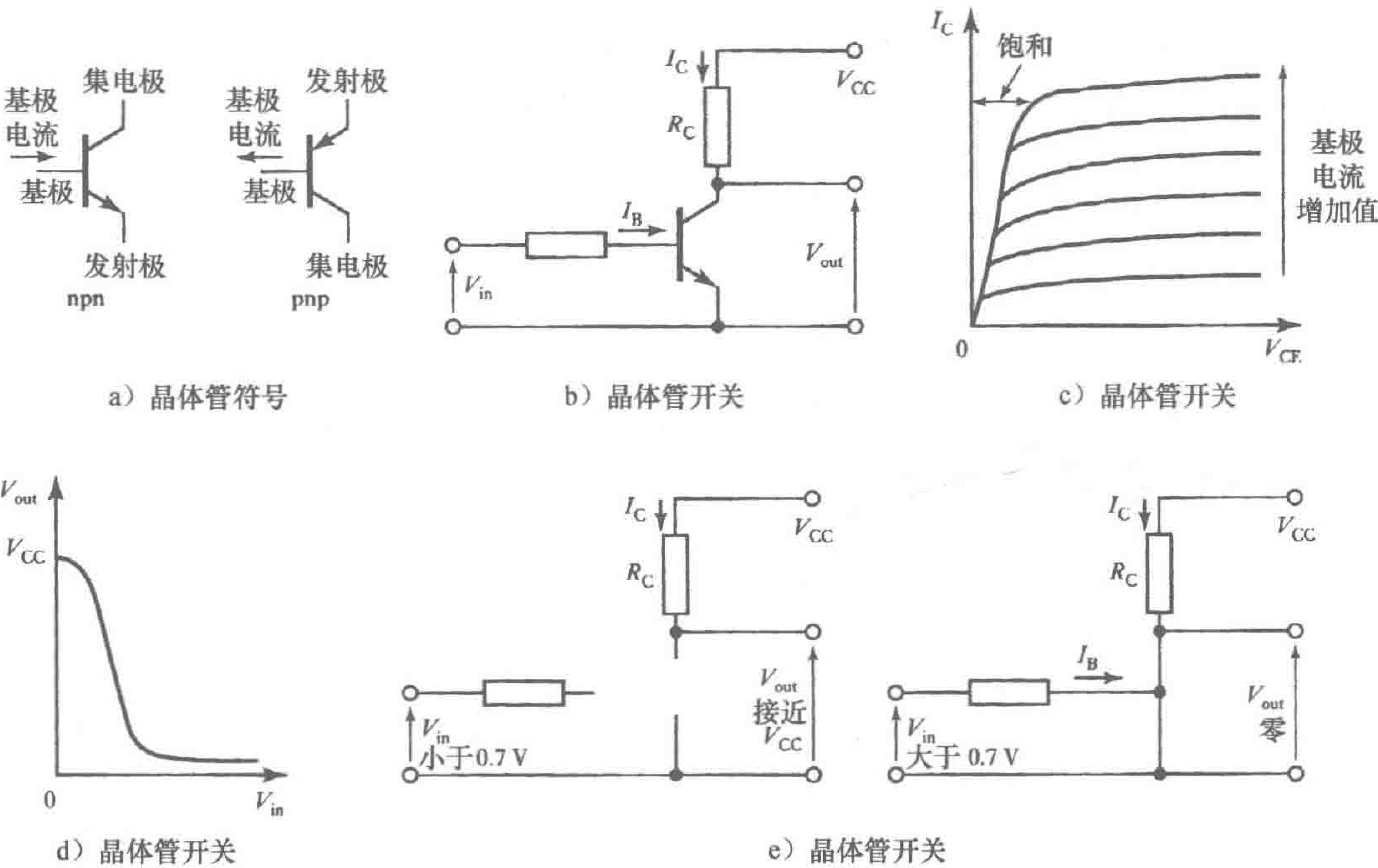


a）相位控制　b）缓冲电路

图 9-8　相位控制与缓冲电路

9.3.3　双极型晶体管

双极型晶体管有 npn 和 pnp 两种类型。图 9-9a 是每种类型的符号。对于 npn 型晶体管，主电流从集电极流入，从发射极流出，将控制信号加载到基极。pnp 型晶体管主电流从发射极流入，从集电极流出，将控制信号加载到基极。



a）晶体管符号　b）晶体管开关　c）晶体管开关

d）晶体管开关　e）晶体管开关

图 9-9　晶体管

图 9-9b 所示的 npn 型晶体管的连接电路，称为共发射极电路。集电极电流 I_C 和集电极与发射极的电势差 V_{CE}的关系，通过图 9-9c 中的一系列曲线来描述。当基极电流 I_B 为 0 时，晶体

管截止。在这种状态下，基极与发射极、基极与集电极都为反向偏置。当基极电流升高时，集电极电流升高并且 V_{CE}减少，使更多的电压加在了电阻 R_C 上。当 V_{CE}达到了 $V_{CE(sat)}$时，基极 - 集电极结处变为正向偏置，即使基极电流继续升高，集电极电流也不再升高，这称为**饱和状态**。当基极电流从 0 到使晶体管趋于饱和的值之间切换时，双极型晶体管可以用做开关。当没有输入电压 V_{in}时，实质上整个电压 V_{CC}都加在了输出端上。当输入电压足够高时，晶体管打开，使 V_{CC}只有非常小的一部分加在了输出端上（见图 9-9d）。图 9-9e 总结了这种典型的晶体管的开关切换行为。

集电极电流 I_C 和基极电流 I_B（晶体管处于未饱和状态时的基极电流）的关系为：

$$I_C = h_{FE} I_B$$

此处 h_{FE}是电流增益系数。在饱和状态集电极电流 $I_{C(sat)}$为：

$$I_{C(sat)} = \frac{V_{CC} - V_{CE(sat)}}{R_C}$$

为了确保晶体管进入饱和状态，基极电流必须至少增加到：

212～213

$$I_{B(sat)} = \frac{I_{C(sat)}}{h_{FE}}$$

因此，当一个晶体管的 $h_{FE} = 50$，$V_{CE(sat)} = 1V$，对一个电路 $R_C = 10\Omega$，$V_{CC} = 5V$，基极电流必须至少为 8mA。

由于驱动一个双极型功率晶体管的基极电流相对较大，所以常常需要第二个晶体管来对微处理器等提供的较小的电流进行转换。此转换电路如图 9-10a 所示。这样的一对晶体管的组合能够将小的输入电流转换为高电流，称为**达林顿管**（或复合晶体管），它们也可以单个芯片的形式得到。因为晶体管经常与电感性负载一起使用会产生很大的瞬时电压，所以常用一个**保护二极管**（续流二极管）与负载反并联，以防止在晶体管关闭的时候被高压损坏。SGS-Thomson 的集成电路 ULN2001 包含 7 个分开的达林顿复合晶体管，每一个都有一个保护二极管。每一个都有额定值为 500mA 的连续工作电流，并且可以承受 600mA 的浪涌电流。

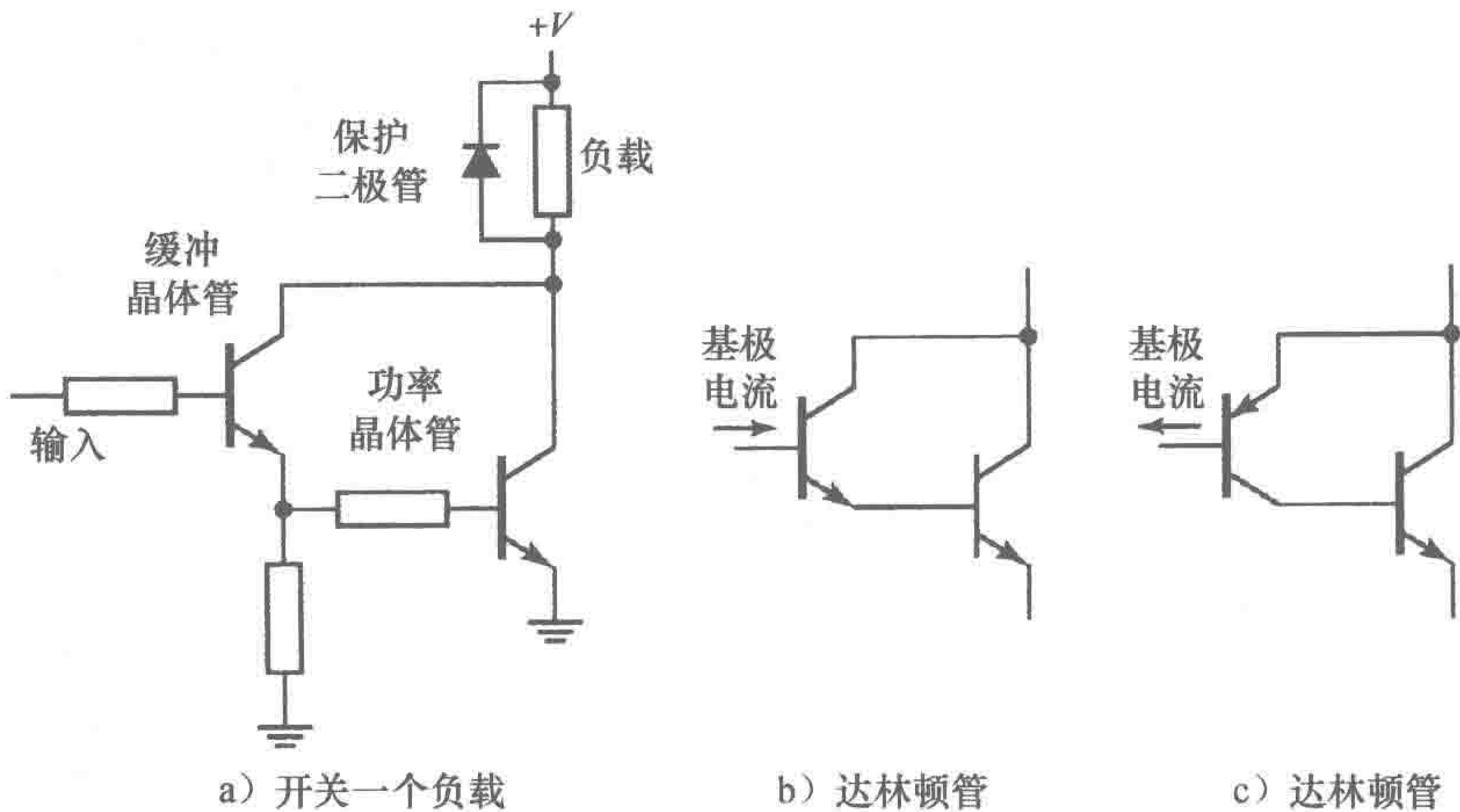


图 9-10　达林顿复合晶体管

图 9-10b 给出了一个小的 npn 晶体管和一个大的 npn 晶体管组合在一起的达林顿连接关系，其结果等效于一个有大的放大系数的大的 npn 型晶体管。图 9-10c 给出了一个小的 pnp 晶体管和一个大的 npn 晶体管组合在一起的达林顿连接关系，结果与一个大的 pnp 晶体管等效。

在使用带有微处理器的晶体管开关驱动电路时，需要注意基极电流的大小和方向。由于要

求的基极电流可能太大，因此会用到一个缓冲器。这个缓冲器把驱动电流提高到了所需要的值。它也可能用来使电路反向。图9-11说明了当使用晶体管来控制一个直流电动机的开关时，怎样来使用缓冲器。型号为240的缓冲器反向，而型号为241和244的缓冲器不反向。缓冲器74LS240的高电平最大输出电流为15mA，低电平最大输出电流为24mA。

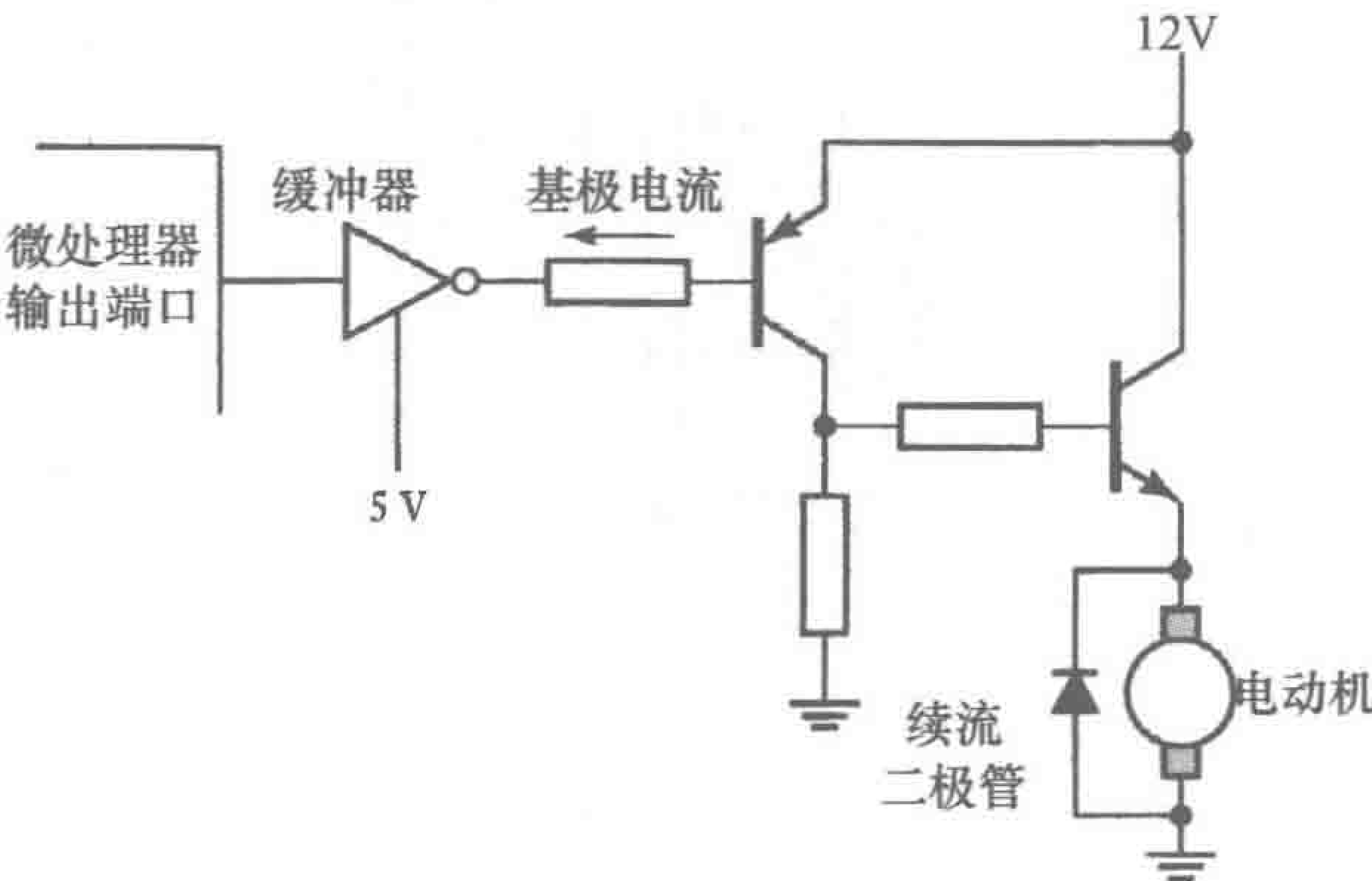


图9-11 直流电动机控制

通过控制基极电流可实现双极型晶体管的开关，并且其开关频率可比晶闸管更高。但是它的功率控制能力比晶闸管弱。

9.3.4 MOSFET

金属氧化物半导体场效应晶体管（Metal-Oxide Field-Effect Transistor，MOSFET）有两种类型，一种是n沟道，另一种是p沟道。图9-12a、b给出了MOSFET的符号。MOSFET与双极型晶体管用于开关控制的主要不同在于MOSFET没有电流流入门来执行控制。其栅极电压是控制信号。没有必要考虑电流的大小，因此驱动电路可以简化。

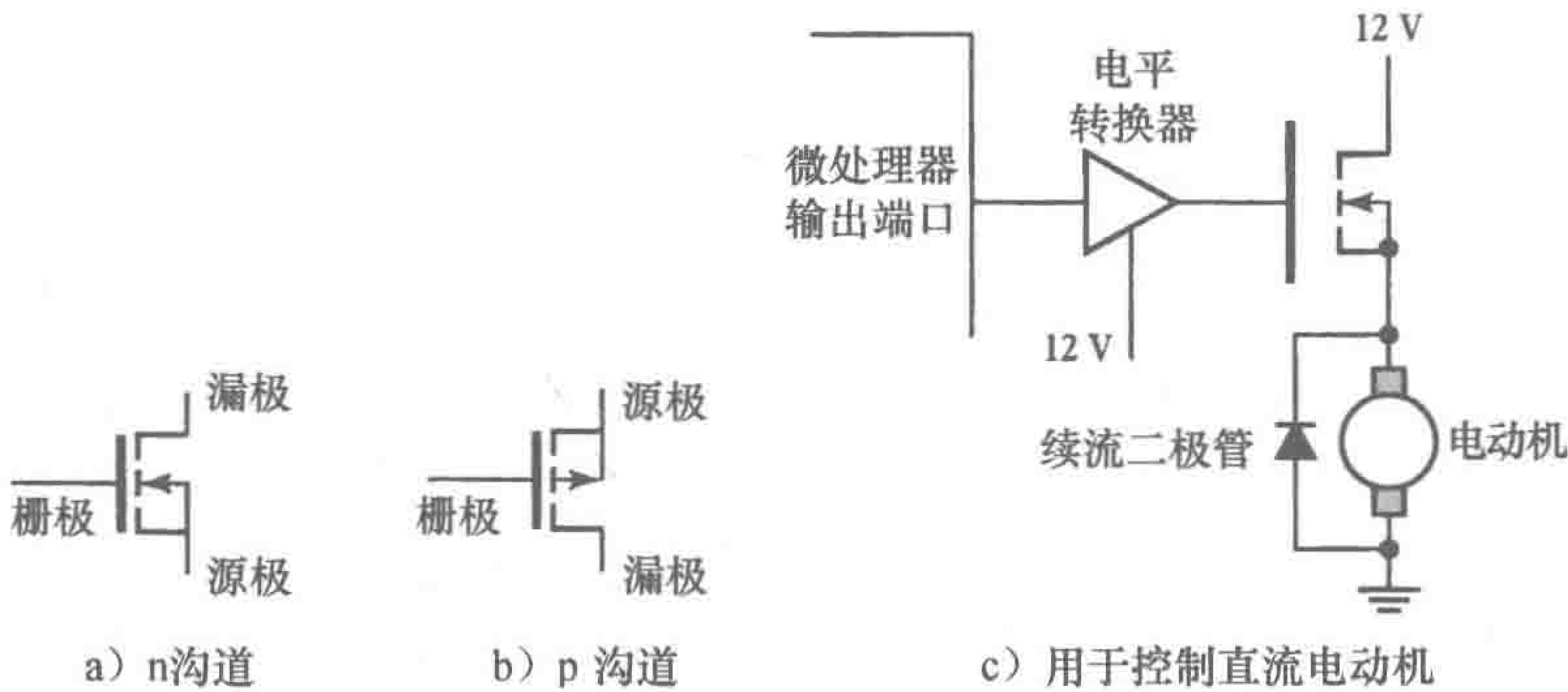


图9-12 MOSFET

图9-12c给出了把MOSFET用做电动机开关的方法。与图9-11中使用的双极型晶体管电路相比，它使用一个电平转换缓冲器，以便提高MOSFET所要求的电压电平。

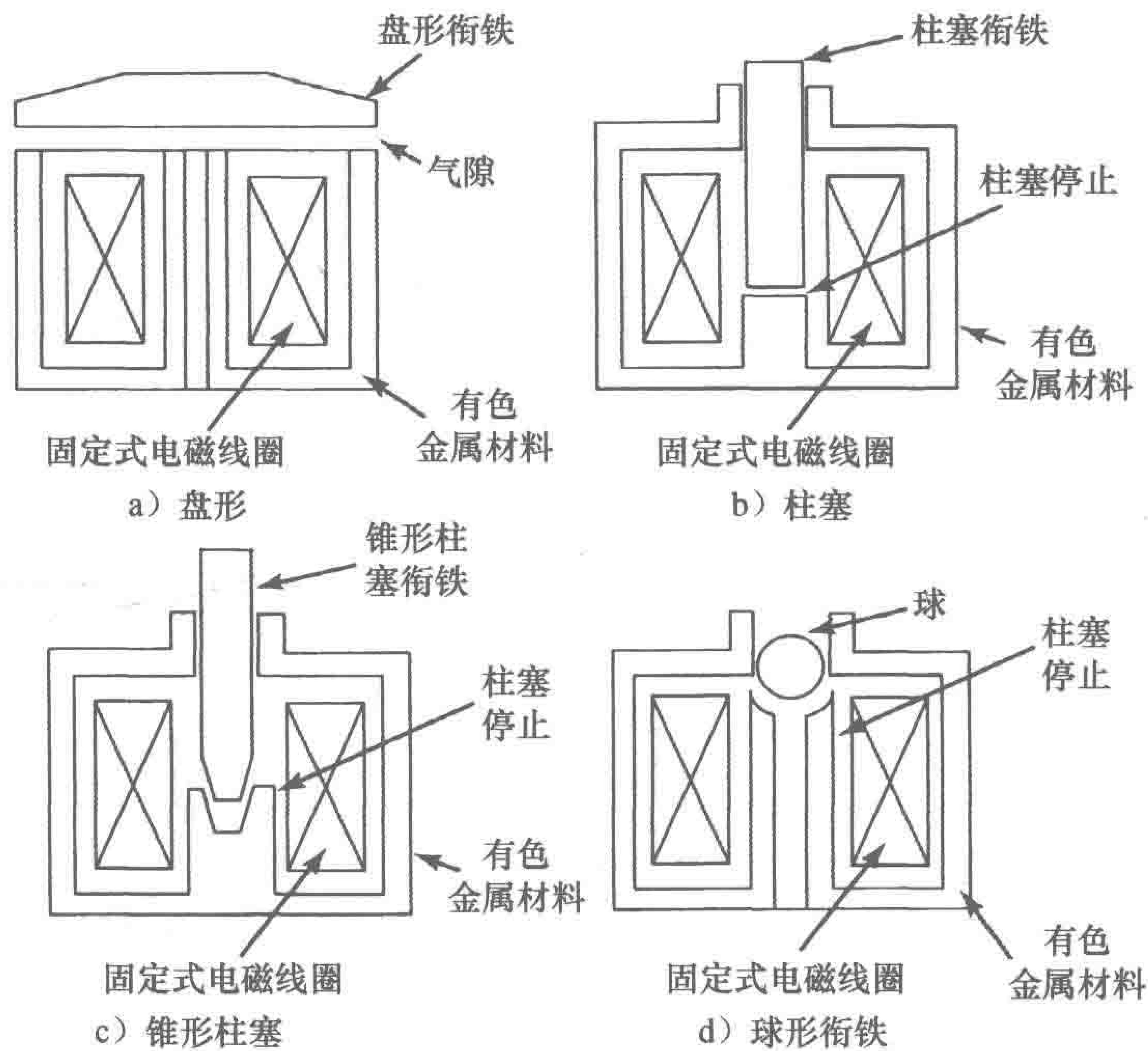
MOSFET使高频转换成为可能，最高可达1MHz。与双极型晶体管相比，它与微处理器的接口更简单。

9.4 螺线管

本质上，螺线管是由线圈和衔铁组成的。当有电流通过线圈时，产生磁场，吸引衔铁。衔铁的运动使复位弹簧收缩，当电流停止时复位弹簧使衔铁回到初始位置。螺线管可以是直线的或旋转的，可以是开/关型或位置可变型的，可在直流或交流下操作。它可用做电动操控的驱动器，广泛应用于短冲程器件中，典型的冲程可达25mm。

214
~
215

图9-13给出了4个线性螺线管的例子，每个螺线管带有不同的衔铁。衔铁、极片和中心管的类型取决于驱动器的用途。在需要短冲程和快速运动时，盘形衔铁是有用的。在需要短冲程和快速运动的应用中，柱塞衔铁也被广泛应用。锥形衔铁应用在长冲程的应用中，典型应用是自动锁门机构。球形衔铁应用于流体控制中，典型应用是气囊展开机构。



a）盘形　b）柱塞　c）锥形柱塞　d）球形衔铁

图 9-13　线性螺线管的基本形式

注：当螺线管中电流停止时，使衔铁返回到其初始位置的弹簧未在图中表示出来。

对一个简单的开关器件，没必要提供线性特性。当需要一个比例式驱动器时，就要精心设计使螺线管电流与衔铁位移成比例。一个简单的使用开关螺线管驱动器的例子是门在螺线管通电的时候被锁上，或者在螺线管通电的时候被打开。

螺线管阀门是这种器件的另一个典型的例子，它用于液压或气动系统中控制流体的流动（见图 7-9）。当电流通过线圈时，一个软铁塞形式的衔铁被拉入线圈中，这样就可以打开或关闭端口允许流体的流动。螺线管施加在衔铁上的力是线圈中的电流和线圈内衔铁的长度的函数。当打开或关闭阀门时，即那些用于方向控制的阀门，控制线圈中的电流要么打开要么关闭，因此衔铁处于这两个位置中的一个。在比例控制阀中，通过控制线圈中的电流使活塞的位移与电流的大小成比例。

螺线管驱动器也可被锁定，即当螺线管中的电流断开时仍保持它们的驱动位置。图 9-14 说明了这一点。增加一个永磁铁，使得当线圈中没有电流通过时也没有足够的力量拉动衔铁克服弹簧达到闭合的位置。然而，当电流流入螺线管产生与永磁铁同方向的磁场时，衔铁被拉到闭合位置。当螺线管中没有电流流过时，永磁铁有足够的吸力使衔铁保持在它的闭合位置上。为了打开它，流经螺线管的电流需要反向，以产生一个与永磁铁磁场方向相反的磁场。因此，这样的螺线管驱动器能用来打开一些装置，并让它一直开着，直到收到相反的电流信号才关闭。

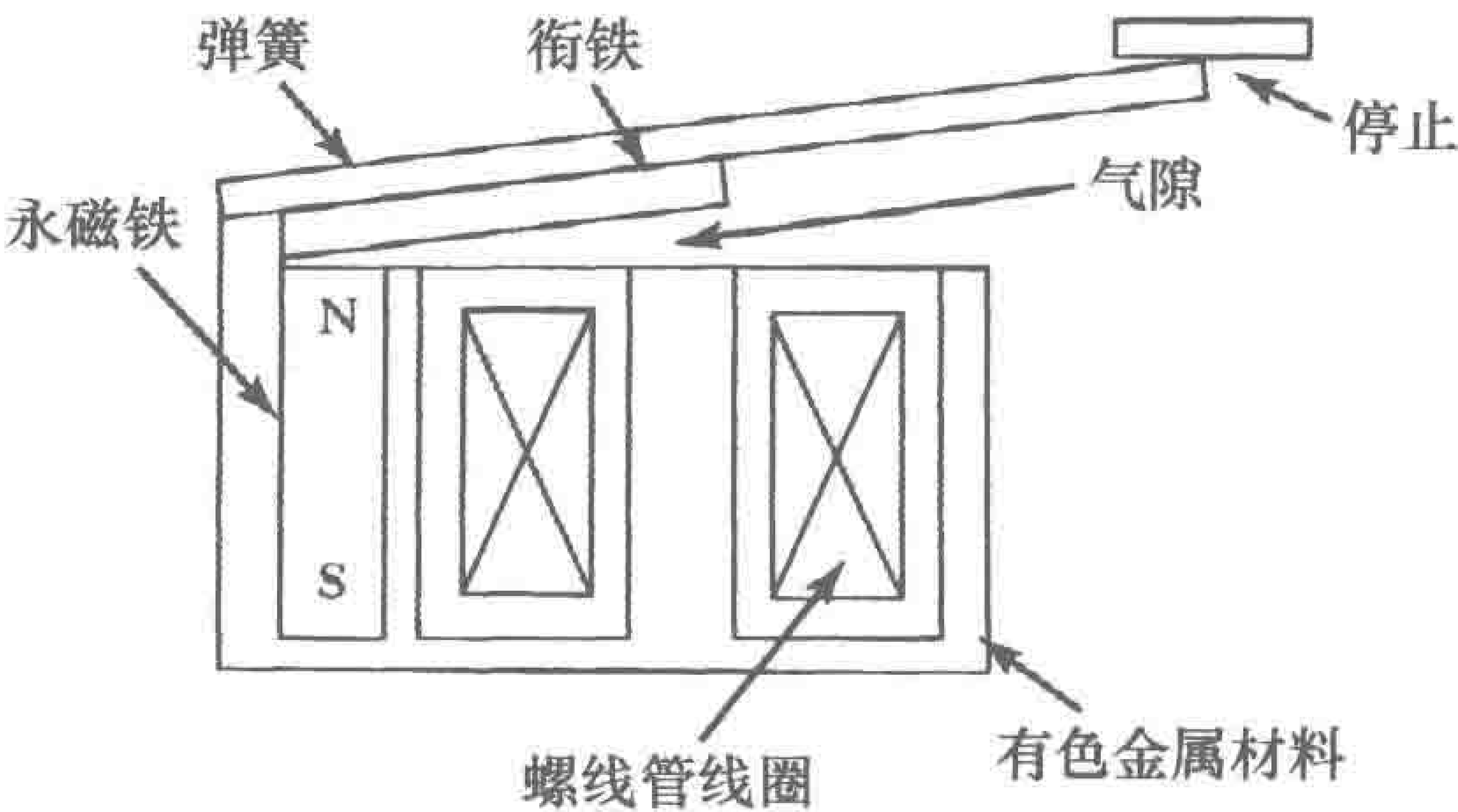


图 9-14　一个锁定螺线管的驱动器

9.5　直流电动机

电动机是位置或速度控制系统中频繁使用的末级控制元件。电动机主要分为两类：直流电

动机和交流电动机。在现代控制系统中用的最多的电动机是直流电动机。直流电动机可以分为两类，一类是使用电刷与装配在转子上的换向环接触，把电流从一个转子绕组上转换到另一个转子绕组上；另一类是无电刷的类型。有电刷的电动机，转子上有线圈绕组，定子可能是一个永磁铁或一个电磁铁。无刷电动机的布置相反，转子是一个永磁铁，而定子上有线圈绕组。

9.5.1　有刷直流电动机

一台**有刷直流电动机**本质上是一个可以自由旋转的线圈，因此称为转子，该转子处在永磁铁或电磁铁产生的磁场中。因为该磁铁是固定的，所以称为定子（见图 9-15a）。当电流通过线圈时，产生的电磁力作用在线圈的边上，产生旋转。然而，为了连续旋转，当线圈通过垂直位置时通过线圈的电流方向必须反向，这是通过电刷与开口换向器的接触来实现的，该换向器和线圈一起旋转。

217

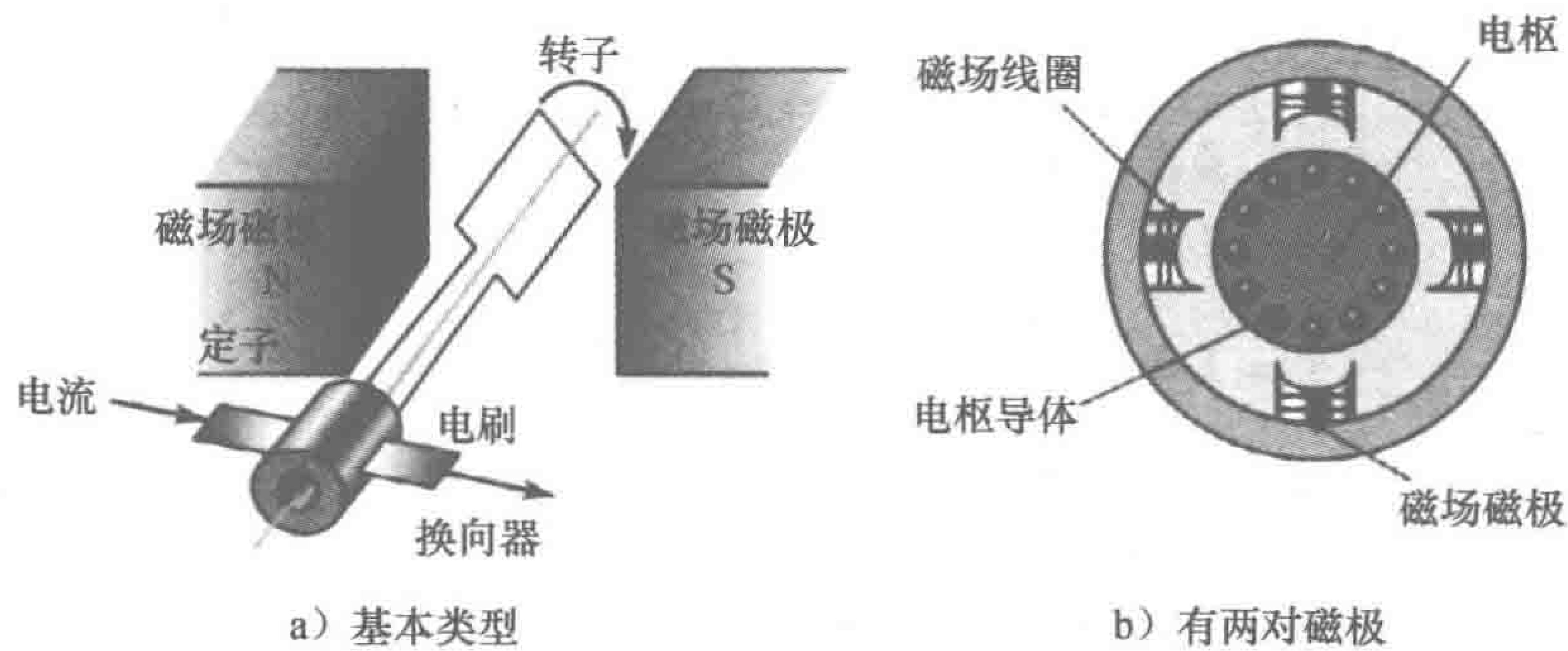


图 9-15　直流电动机

在传统的直流电动机上，线圈被安装在一个称为**电枢**的磁性材料圆柱体的沟槽里。电枢安装在轴承上并能自由旋转。它安装在**励磁极**产生的磁场中。对小电动机来说，这些可能是永磁体或拥有磁性的电磁铁，磁性由通过**励磁线圈**的电流产生。图 9-15b 给出了一个 4 极直流电动机和由载流线圈产生的磁场的基本原理。每个电枢线圈的末端和换向器（分段环形物）相连。同时，电刷（碳触体）与换向器发生电接触。电枢转动，当它转动到磁场两极之间时换向器将每个线圈中的电流换向。如果作用在线圈上的力保持作用方向相同，这样的转动才可持续。通过翻转电枢电流或磁场电流的方向，可以改变直流电动机的转动方向。

考虑一台永磁直流电动机，永磁体给出了一个恒定的磁感应强度值。对于长 L 的电枢导体通过带电流 i，由磁感应强度 B 产生的垂直于导体的力的大小是 BiL（见图 9-16a）。该力产生了关于 Fb 的线圈轴线的转矩 T，其中 b 是线圈的宽度。因此：

$$\text{电枢圈数为 1 的转矩 } T = BbLi = \Phi i$$

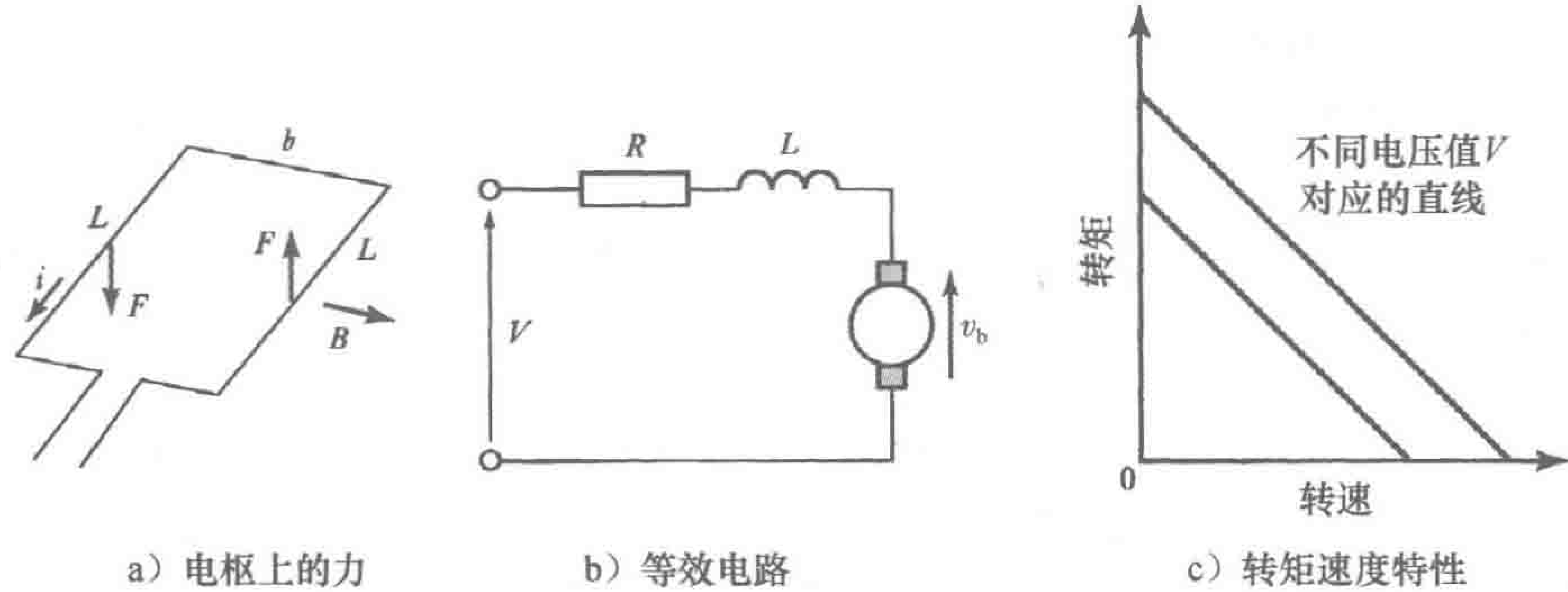


图 9-16　直流电动机

218

其中 Φ 是每圈电枢内的磁通量。实际上线圈圈数超过一个，并且磁极超过一对，所以我们可以写成：

$$\text{转矩 } T = k_t \Phi i$$

k_t 是常量。等式也可写成 $T = K_t i$，其中 K_t 称为电动机的转矩常量。由于电枢线圈在磁场中旋转，所以电磁感应现象将会出现，并且产生一个反电动势。反电动势大小 v_b 与线圈磁通变化率成正比，并且对于一个恒定磁场，其正比于旋转的角速率 ω。因此：

$$\text{反电动势 } v_b = k_v \Phi \omega$$

其中 k_v 是常量。此等式也可写成 $v_b = K_v \omega$，其中 K_v 是电动机的反电动势常量。

我们可以把直流电动机看成是图 9-16b 的等效电路，即电枢线圈等效于电阻 R 和电感 L、反电动势 v_b 串联。如果我们忽略电枢线圈的电感，则电阻上的电压为 V 减去反电动势，即 $V - v_b$。因此，通过电阻的电流为：

$$i = \frac{V - v_b}{R} = \frac{V - k_v \Phi \omega}{R} = \frac{V - K_v \omega}{R}$$

转矩 T 为：

$$T = k_t \Phi i = \frac{k_t \Phi}{R}(V - k_v \Phi \omega) = \frac{K_v}{R}(V - K_v \omega)$$

角速度 ω 对应的转矩曲线如图 9-16c 所示，每条直线对应不同的电压值。启动转矩，即当 $\omega = 0$ 时，代入上述导出的公式，得出 $K_t V/R$，因此正比于加载的电压和启动电流 V/R。随着速度提高，转矩下降。如果一台永磁电动机产生 6Nm 的转矩和 2A 的电枢电流，根据 $T = K_t i$，由 1A 电流得到的转矩为 3Nm。

永磁电动机的速度取决于通过电枢的电流，因此可以通过改变电流调节速度。当电动机在稳定状态下工作时，由电功率转换的机械功率是由转矩和角速度的乘积。在稳定状态下传递给电动机的功率是通过电枢线圈电阻的功率损失和输出的机械功率的和。

举个例子，一个小的永磁电动机 S6M41（PMI Motors）的参数为：$K_t = 3.01$N cm/A，$K_v = 3.15$V 每千转/分，一个终端电阻为 1.207Ω，一个电枢电阻为 0.940Ω。

219

9.5.2　含励磁线圈的有刷直流电动机

含励磁线圈的直流电动机根据励磁线圈和电枢线圈的连接方式可分为：**串励**（series）、**并励**（shunt）、**复励**（compound）和**他励**（separately excited）四类。

1. 串励电动机

在串励电动机（见图 9-17a）中，电枢和励磁线圈串联，它们的电流相等。磁通量 Φ 取决于电枢电流 i_a，于是电枢上的转矩为 $k_t \Phi i_a = k i_a^2$。启动时 $\omega = 0$，$i_a = V/R$，启动转矩 $= k(V/R)^2$。由于这类电动机电阻低，所以有很高的启动转矩和空载速度。随着速度增加，转矩降低。由于 Ri 很小，所以 $V = v_b + Ri \approx v_b$，并且由于 $v_b = k_v \Phi \omega$ 和 Φ 正比于 i，所以有 V 正比于 $i\omega$。合理的近似值 V 是常量，速度反比于电流，当负载增加时速度下降。颠倒线圈的两极对电动机旋转的方向没有影响。它将继续沿着相同方向旋转，这是由于磁场和电枢电流的方向同时被颠倒了。这样的直流电动机用于需要大的启动转矩的场合。在轻负载时，串励电动机会有危险，因为电动机的速度可能会过高。

2. 并励电动机

在并励电动机（见图 9-17b）中，电枢和励磁线圈并联。它提供了最低的启动转矩、一个相对很低的空载速度和较好的速度调节性能。励磁线圈由许多匝细金属丝绕成，因此有比电枢线圈大得多的电阻。因此，在一个恒定的供电电压下，励磁电流实质上是恒定的。启动转矩为 $k_t V/R$，因此它提供了一个低启动转矩和低空载速度。由于 V 几乎恒定不变，所以无论负载怎

样，电动机提供的速度都几乎是恒定的，并且由于这个特性此类电动机被广泛应用。为了颠倒旋转方向，电枢或者磁场需被反向。

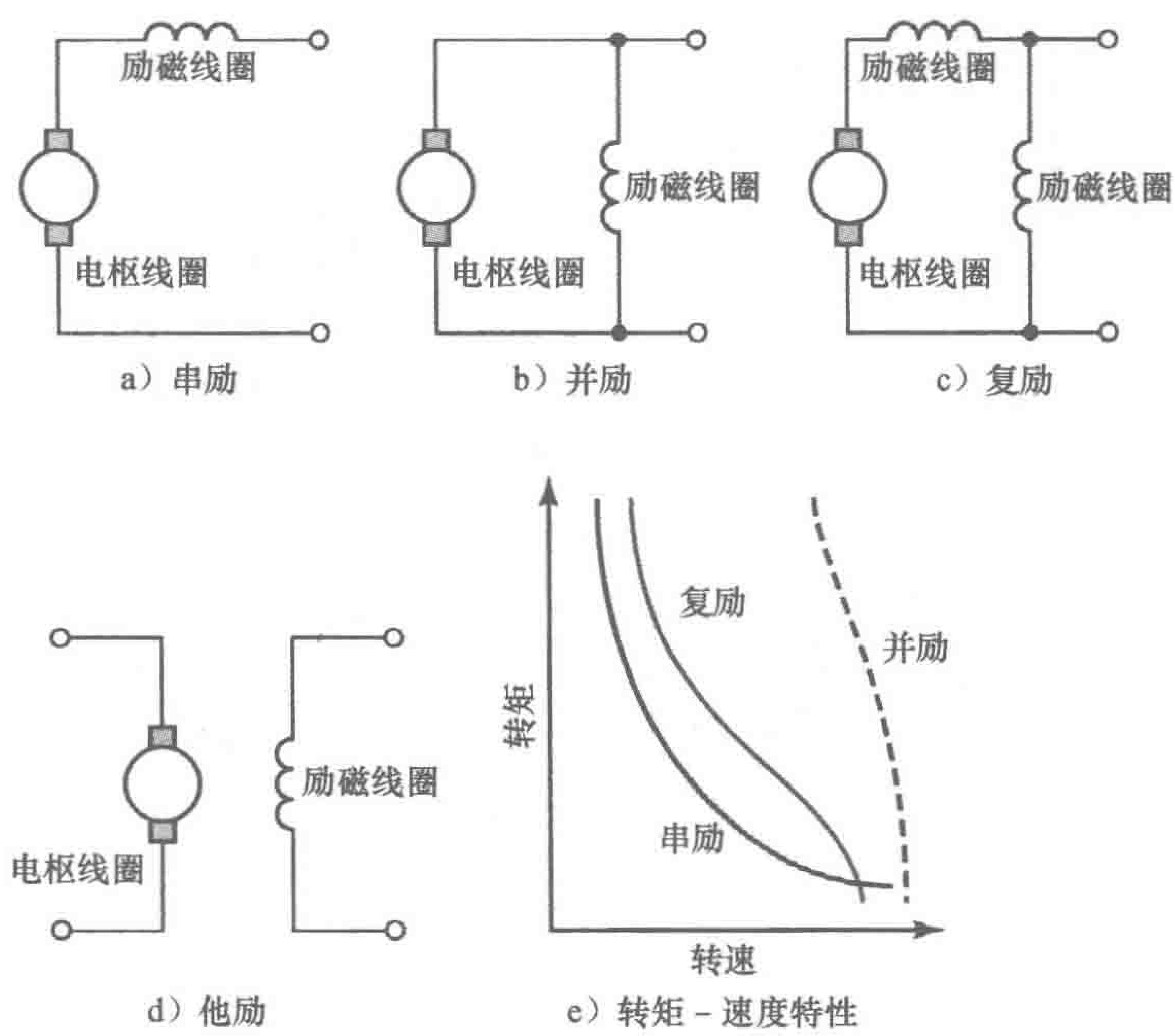


图 9-17　直流电动机

220

3. 复励电动机

复励电动机（见图 9-17c）有两个励磁线圈，一个和电枢串联，另一个和电枢并联。目的是获取串励电动机和并励电动机的优点，也就是一个高的启动转矩和好的速度调节性能。

4. 他励电动机

他励电动机（见图 9-17d）分开控制电枢和励磁电流，并且可以认为是并励电动机的一种特殊情况。

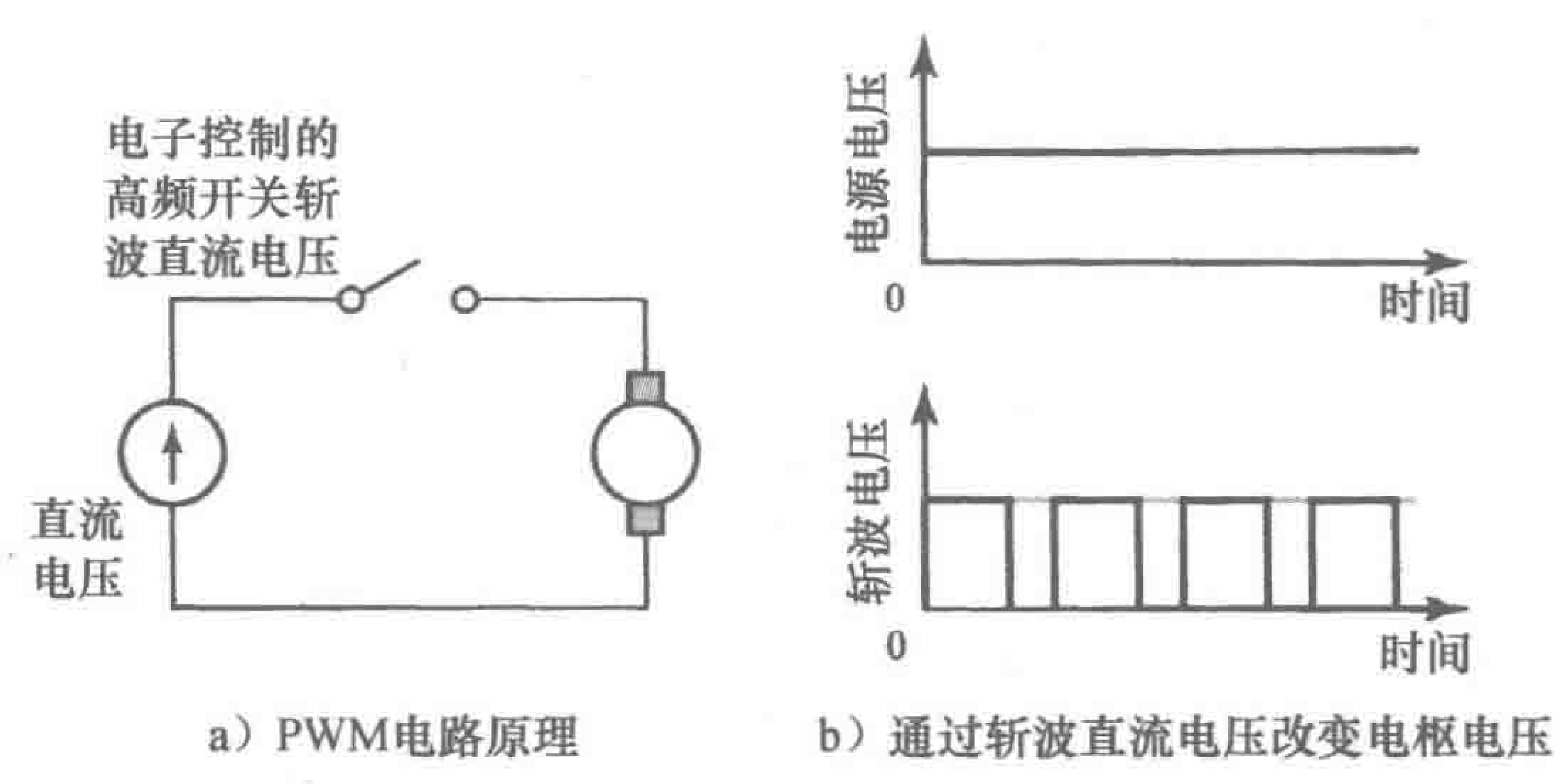
这样的直流电动机的速度可以通过改变电枢电流或励磁电流来改变。通常改变的是电枢电流。这可以通过串联电阻来实现。然而，这种方法是非常低效的，因为控制器电阻消耗了大量的电能。另一个可选的方法是控制电枢电压（见 9.5.3 节）。直流电动机在静止状态下产生一个转矩来实现自启动。然而，由于电流 $i=(V-v_b)/R$，所以电动机需要一个初始电阻去限制启动电流。由于启动时没有反向电动势来限制电流，所以启动电流可能会很大。

电动机的选择取决于具体的应用。以机器人的操作臂为例，机器人的腕关节可能需要一个串励电动机，因为随着负载增加，速度降低。并励电动机将用在需要恒定速度而不考虑负载的地方。

9.5.3　有刷直流电动机的控制

永磁电动机的速度取决于电枢线圈的电流。励磁电动机的速度可通过改变电枢电流或励磁电流来改变。通常采用改变电枢电流的方法。因此，通过控制加载到电枢上的电压来获得速度的控制。然而，因为电源电压通常是固定的，所以可通过电子电路来实现电压变化。

在提供交流电压时，使用图 9-4b 的晶闸管电路来控制加载到电枢上的平均电压。然而，我们通常关心如何通过从微处理器发出控制信号来控制直流电动机。在这种情况下，通常应用**脉冲宽度调制**（PWM）技术。这主要是对恒定的直流电压进行斩波，因此电压的平均值就会被改变（见图 9-18）。



a）PWM电路原理　　b）通过斩波直流电压改变电枢电压

图 9-18　PWM

图 9-19a 给出了怎样通过一个基本的晶体管电路获得 PWM。晶体管通过一个加载到它的基极上的信号接通或断开。当晶体管截止时，电动机变成了一个发电机，电流上升，二极管给电流提供一个通路。这样的电路只能用于驱动电动机单向转动。一个包含 4 个晶体管的电路（见图 9-19b），称为 H 电路，能使电动机正反两个方向旋转。这个电路可以通过使用逻辑门来改进，一个输入控制电动机的开关，一个控制旋转方向（见图 9-19c）。

a）基本晶体管电路

b）H型电路

c）有逻辑门的H型电路

图 9-19　H 型电路

221
~
222

以上都是开环控制的例子。这建立在条件保持不变的假设上，例如恒定的电压和电动机负载。闭环控制系统在条件改变时使用反馈来修正电动机速度。图 9-20 给出了一些可能被采用的方法。

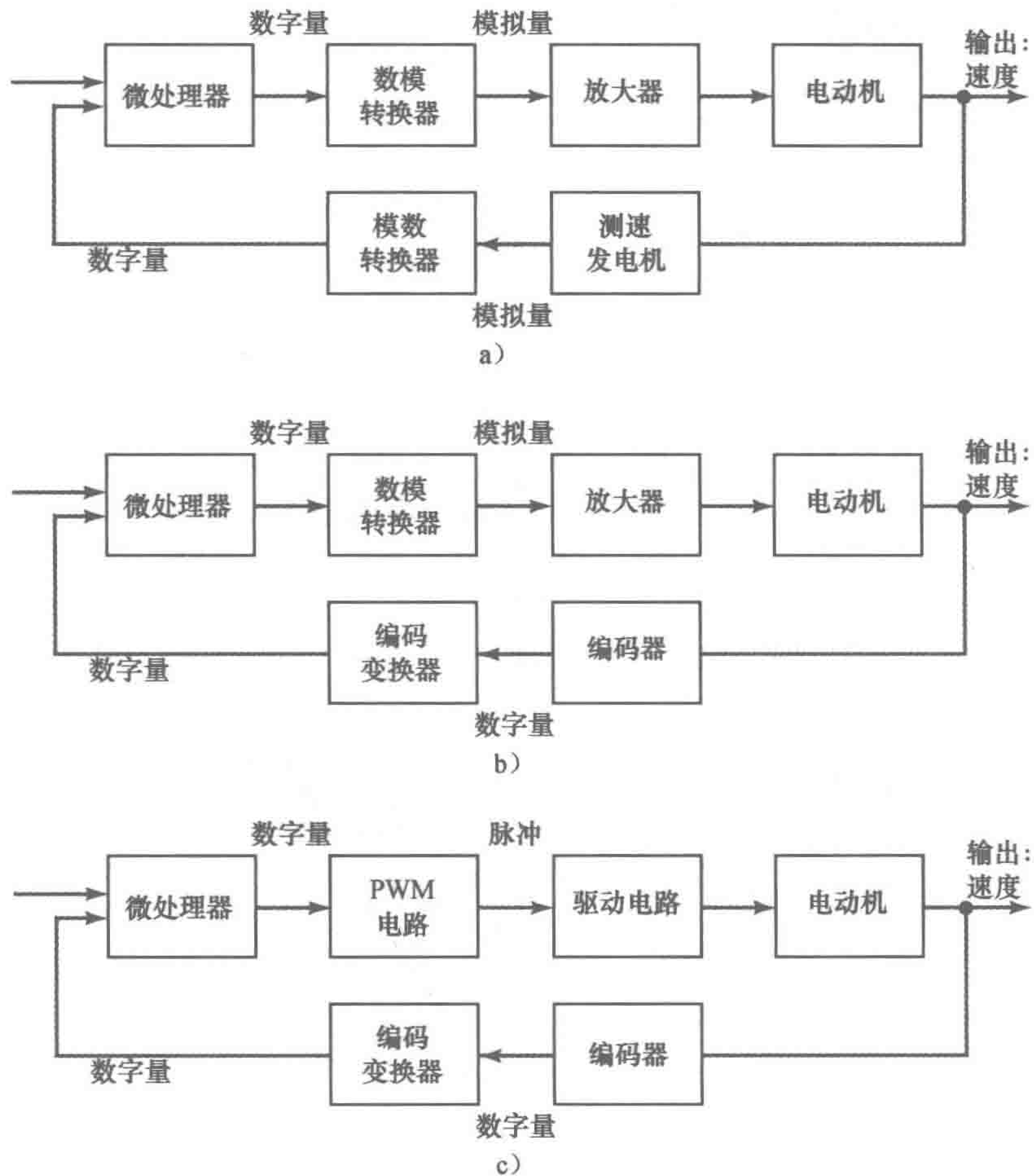


图 9-20　速度反馈控制

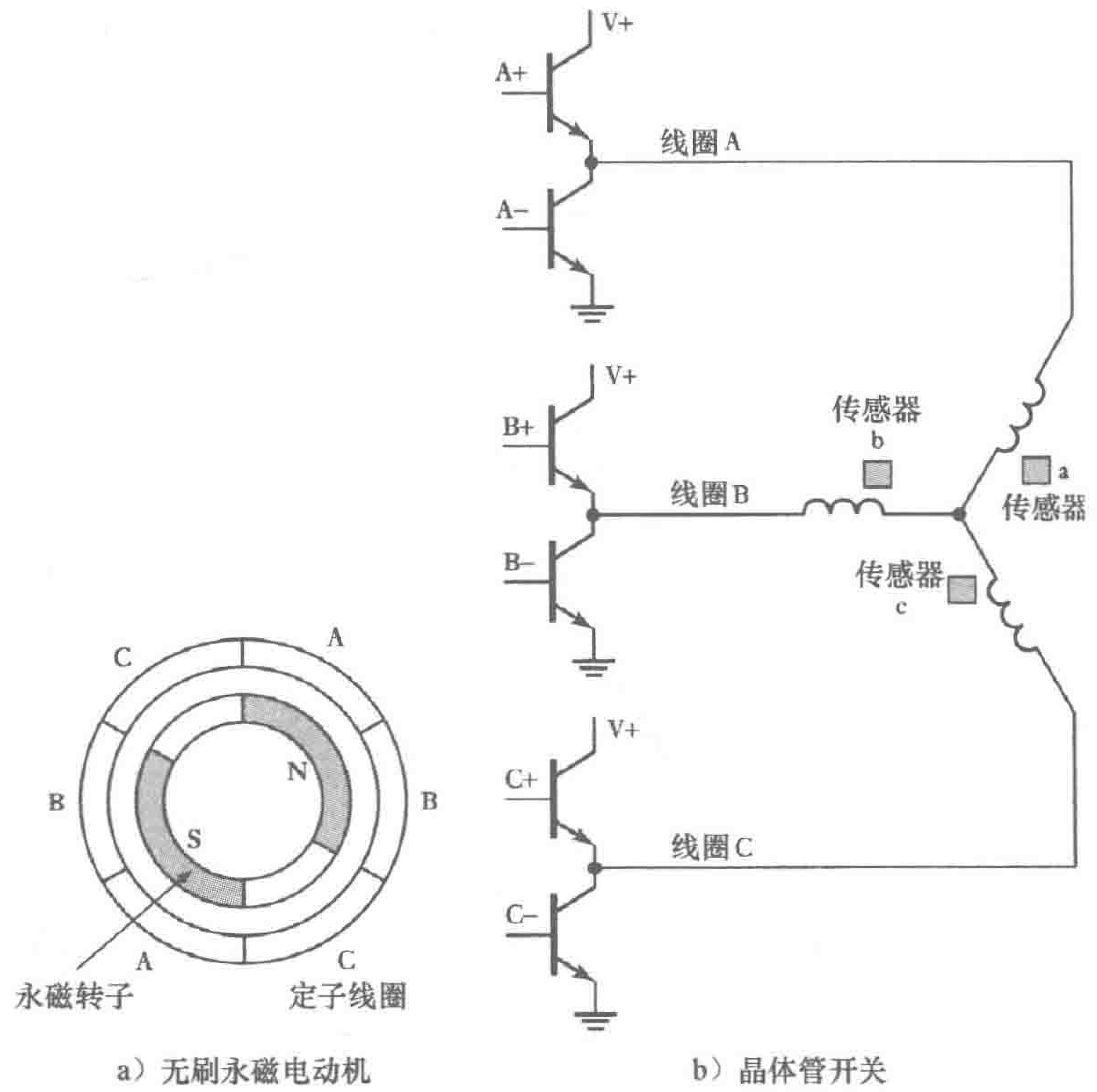
在图 9-20a 中的反馈信号由测速发电机提供，它给出模拟信号，通过 ADC 转换成数字信号作为微处理器的输入。微处理器的输出通过 DAC 转化为模拟信号，改变加载到直流电动机电枢上的电压。在图 9-20b 中的反馈信号由编码器提供，它给出数字信号，经过编码转换后可直接输入到微处理器中。如图 9-20a 所示，系统输出了一个已改变的模拟电压来控制电动机的速度。在图 9-20c 中系统完全是数字化的，PWM 用来控制加载到电枢上的平均电压。 223

9.5.4　无刷永磁直流电动机

有刷直流电动机的一个问题是，它们需要换向器和电刷来周期性地改变通过每个电枢线圈的电流方向。电刷和换向器滑动接触，因此两者之间会产生火花，并且受到磨损。因此电刷需要定期更换，并且换向器也要用新表面覆盖。为了避免这个问题，**无刷电动机**被设计出来了。

基本上它们由一系列的定子线圈和一个永磁转子组成。磁场中的载流导体会受到力的作用。类似地，根据牛顿第三运动定律，磁体也受到一个大小相等、方向相反的力。在常规的直流电动机中，磁体是固定的，载流导体运动。在无刷永磁直流电动机中，情况则相反：载流导体是

固定的，而磁体运动。转子是铁氧体或陶瓷永磁体。图 9-21a 给出了这样一个电动机的基本样式。定子线圈的电流以绕着线圈的顺序通过晶体管进行电子切换，这个切换由转子的位置控制，于是总是有作用在磁体上的力使它朝着相同方向旋转。放在定子周围的霍尔传感器通常用来感应转子的位置，并触发晶体管进行电流切换。



a）无刷永磁电动机　　b）晶体管开关

224

图 9-21　无刷电动机

图 9-21b 给出了可能和图 9-21a 的电动机一起使用的晶体管切换电路。为了按顺序切换线圈，我们需要按正确顺序提供信号来接通晶体管。信号由译码电路处理后的 3 个传感器的输出来提供，给出正确的基极电流。因此，当转子在垂直位置时，如 0°，有一个来自传感器 c 的输出信号，没有来自 a 或 b 的信号。这用来接通晶体管 A + 和 B − 。当转子在 60°的位置时，有来自传感器 b 和 c 的信号并且晶体管 A + 和 C − 被接通。表 9-1 给出了整个切换顺序。控制这样的电动机的电路可在单个集成电路中实现。

表 9-1　切换顺序

转子位置	传感器信号			晶体管打开	
	a	b	c		
0°	0	0	1	A +	B −
60°	0	1	1	A +	C −
120°	0	1	0	B +	C −
180°	1	1	0	B +	A −
240°	1	0	0	C +	A −
360°	1	0	1	C +	B −

无刷永磁直流电动机在高性能、高稳定性和低维修性的场合越来越得到应用。因为它们没有电刷，所以它们更安静并且能够高速旋转。

9.6　交流电动机

交流电动机可以分为两类：单相和多相的，每一类进一步分成感应电动机和同步电动机。单相电动机用在低功率的场合，而多相电动机用在有高功率要求的场合。感应电动机比同步电动机便宜，因此得到更广泛的应用。

单相笼型感应电动机由一个鼠笼型转子构成，由铜或铝棒插入两端短路环的插槽以形成完整的电路（见图9-22a）。外部没有电路连到转子上。基本的电动机由转子和一个有一套线圈的定子构成。当交变电流流过定子线圈时，产生交变磁场。由于存在电磁感应，所以转子的导体中产生感应电动势并且有电流流进转子。最初，当转子静止时，作用在定子磁场中转子上的载流导体的力没有产生净转矩。电动机不会自我启动。很多方法可以用来提供这个原动力，使电动机自启动。一种方法是使用一个辅助的启动线圈来给电动机一个初始推力。转子转动的速度取决于作用在定子上的交变电流的频率。给双极单相电动机提供一个恒定频率的电源，磁场也会以这个频率旋转。磁场的转动速度称为**同步转速**。转子不会完全匹配这个转动频率，一般与其相差1%～3%。这种差别被称为**滑差**（slip）。因此对于一个50Hz频率的电源，电动机转动的速度大概是50转/秒。

225

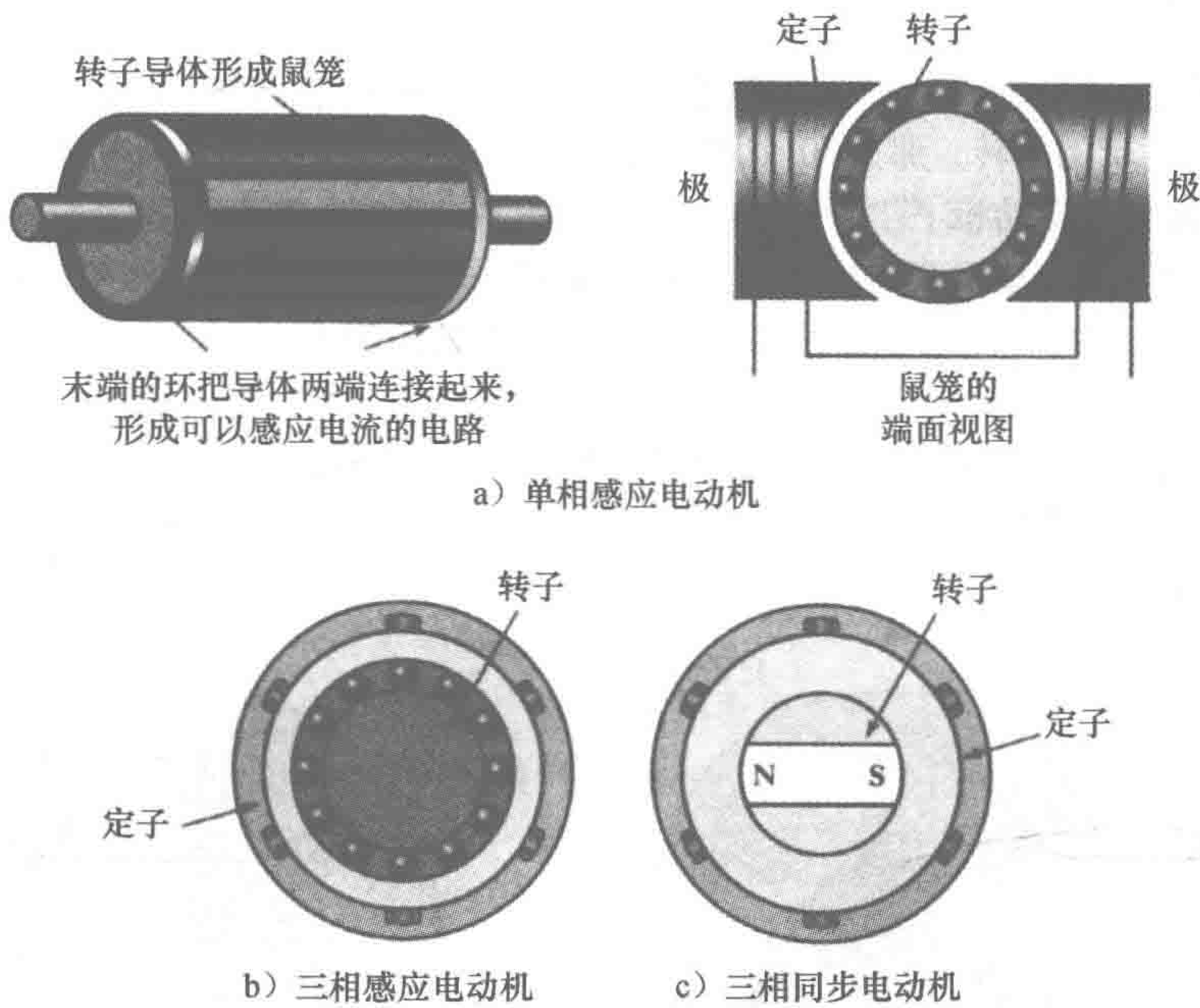


图9-22　交流电动机

三相感应电动机（见图9-22b）与单相感应电动机相似，只是前者的定子带有三个互成120°的线圈。每一个线圈连到三条供电线中的一条。因为三相在不同时间达到它们的最大电流，所以磁场可以认为是绕着定子极旋转，在电流的一个完整周期内完成一次旋转。三相磁场的转动比单相更平滑。三相电动机相比于单相电动机的显著优势在于三相电动机能够自启动。旋转方向可通过交换任意两根线的连接来改变，因为这改变了磁场旋转的方向。

同步电动机有与以上描述的感应电动机相似的定子，但是它有一个永磁转子（见图9-22c）。定子绕组接通交流电会产生旋转磁场，因此磁体跟着它旋转。在每相供电的一对磁极中，在一

个供电周期内，磁场旋转 360°，因此这种布置下旋转频率和供电频率一样。同步电动机用于需要精确速度的场合。它们不能自启动，需要一些系统加载到上面帮助启动它们。

交流电动机在更便宜、更坚固、可靠和免维护方面比直流电动机有巨大优势。然而，交流电动机的速度控制一般比直流电动机更复杂，因此一个速度可控的直流驱动器比交流驱动器更便宜。由于技术的发展和固体元器件价格的下降，价格差别正在缩小。交流电动机的速度控制依赖于可变频率供电，因为此类电动机的速度由供电的频率决定。当加载的定子电压频率比固定时，交流电动机的转矩也固定。因此当频率可变时，为了在不同转速下保持一个恒定的转矩，加载到定子上的电压也必须跟着改变。一种方法，交流电流一开始通过**整流器**被整流为直流电流，然后再**逆变**为一个频率可选择的交流电流（见图 9-23）。另一种常用来操作低速电动机的方法是用**交交变频器**。这种方法直接把一种频率的交流电流转换为另一种频率的交流电流，不需要中间过渡的直流转换。

226

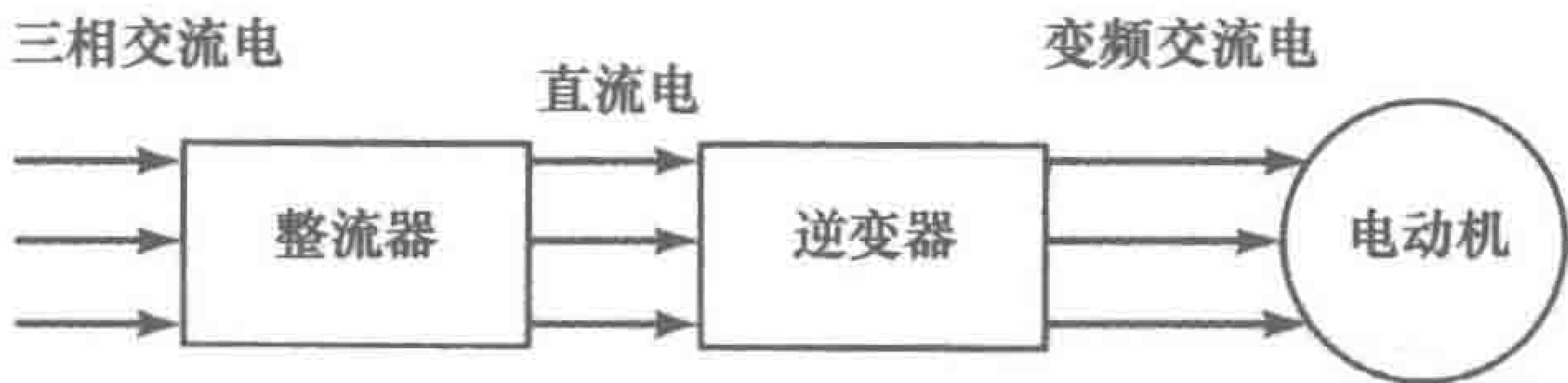


图 9-23　调速交流电动机

9.7　步进电动机

步进电动机是一种对每一个数字脉冲输入产生相等角度步进旋转的设备。因此，假如这种电动机一个脉冲产生 6°的旋转，那么 60 个脉冲将产生 360°的旋转。步进电动机有很多不同的种类：

1. 可变磁阻步进电动机

图 9-24 给出了可变磁阻步进电动机的基本形式。在这种形式下，转子由软铁心构成，呈圆柱形，并带有 4 个极（比定子上的极数少）。当相对的线圈有电流时，磁场随着磁力线产生，磁力线从定子极穿过最近的转子极。由于磁力线可以看作橡皮筋，并且总是试着缩短自己，所以使转子移动，直到和定子排成一行。这时位置称为最小磁阻位置。这种形式的步进电动机的步进角度通常为 7.5°或 15°。

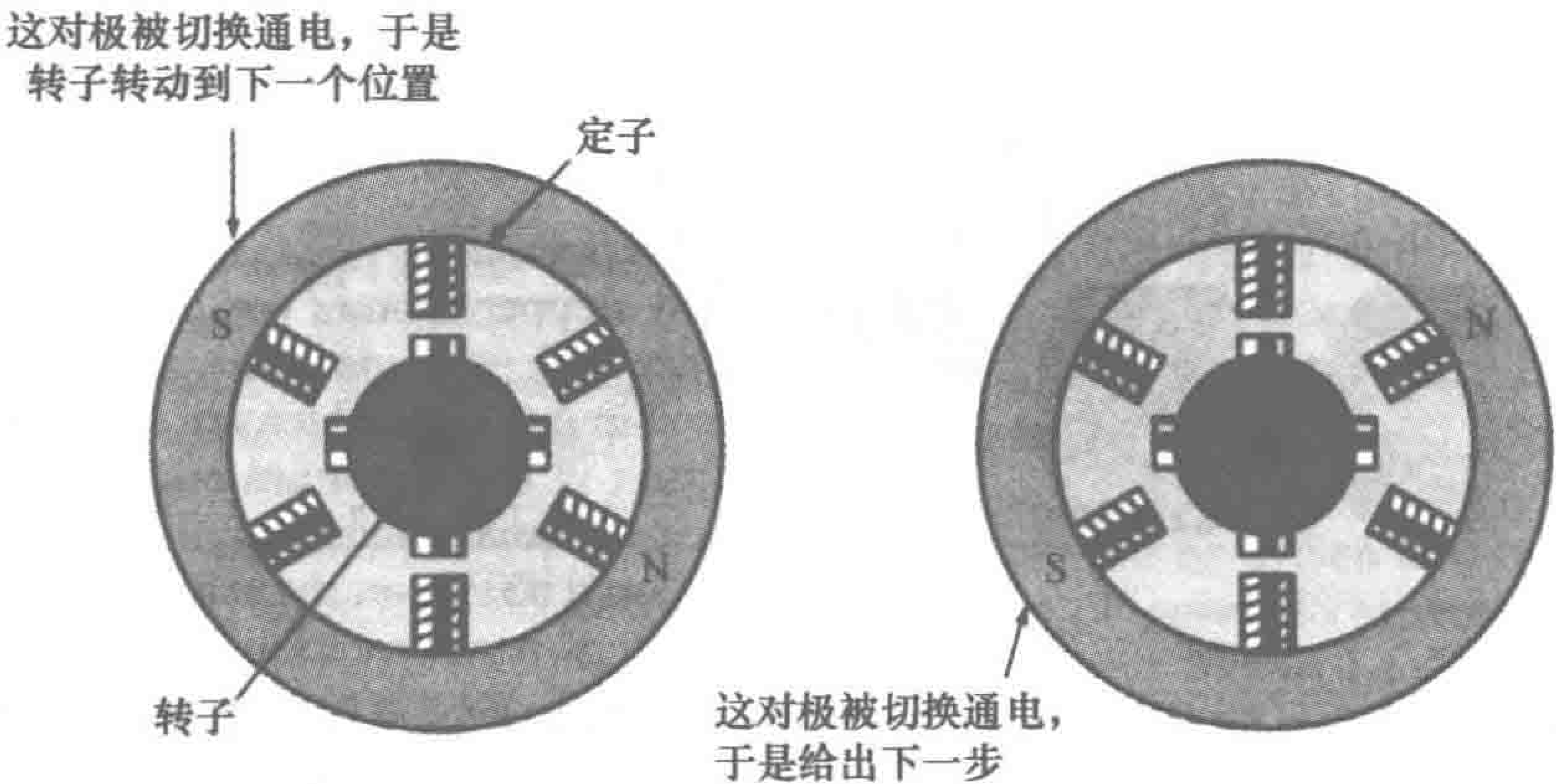


图 9-24　可变磁阻步进电动机

2. 永磁步进电动机

图 9-25 给出了永磁电动机的基本形式。这种电动机有一个带有 4 个极的定子。每一个极上都缠绕有励磁绕组，相对极的线圈串联。通过开关，直流电源向线圈提供电流。转子是一个永磁体，因此当一对定子有电流时，转子将转动与之对齐。因此对图中给定的电流，转子转到 90°的

位置上。如果电流接着切换，极性颠倒，转子会再转 90°去对齐。因此通过在线圈之间切换电流，转子 90°步进。对于这种类型的电动机，步进角度通常是 1.8°、7.5°、15°、30°、45°或 90°。

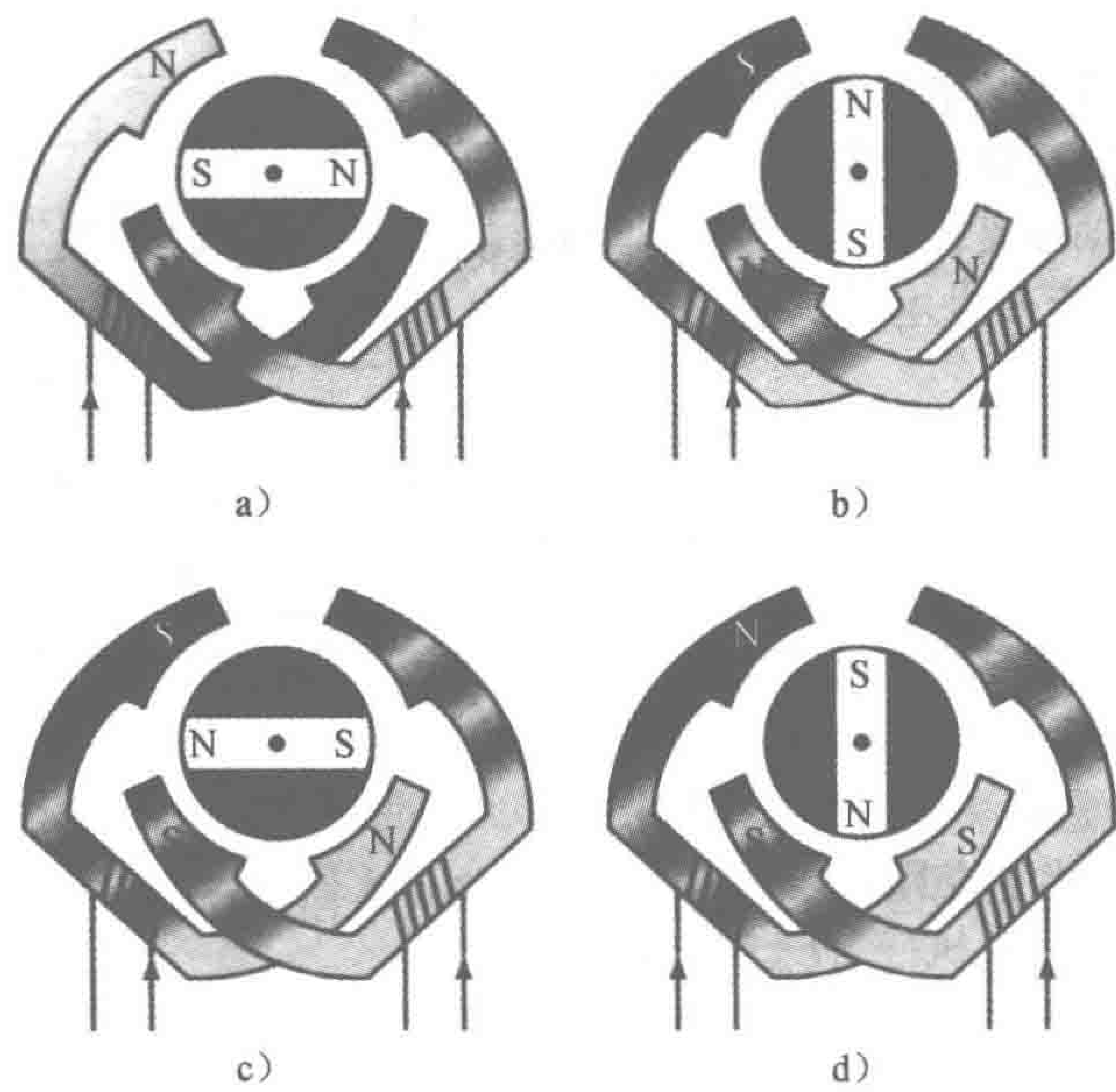


图 9-25　90°步进的两相永磁步进电动机。a)、b)、c) 和 d) 显示了当线圈在不同的方向通电时磁转子的位置

3. 混合步进电动机

混合步进电动机综合了可变磁阻步进电动机和永磁电动机的特性；有一个永磁体装在两个铁盖之间，铁盖上切出了齿（见图 9-26）。对于一对通电的定子线圈，转子转到磁阻最小的位置。典型的步进角度是 0.9°和 1.8°。如果电动机定子上有 n 相，转子上有 m 个齿，则每一圈总的步进数是 nm。这样的步进电动机广泛地应用于高精度定位中，例如计算机硬盘的驱动。

227
~
228

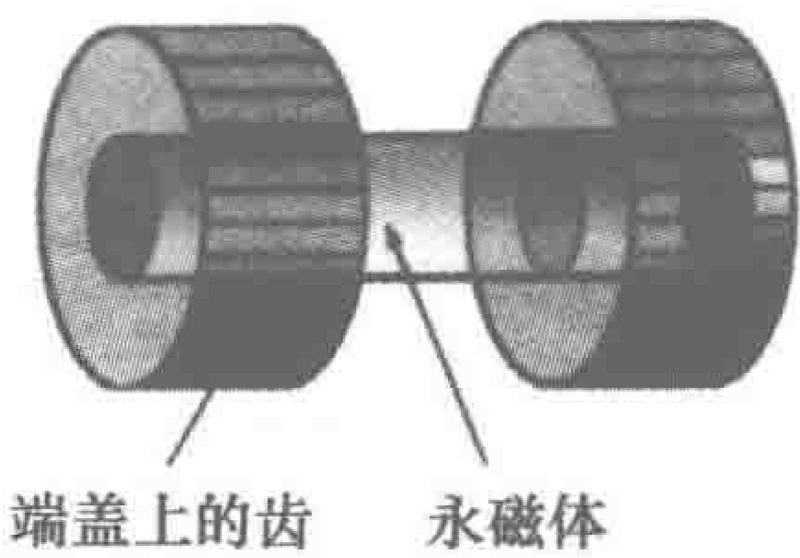


图 9-26　混合步进电动机转子

9.7.1　步进电动机技术参数

下面是一些常常用于说明步进电动机的术语。

1. 相

这个术语是指定子上独立的线圈数，例如四相电动机。指定每相需要的电流和它的电阻、电感，这样也就确定了控制器的开关量输出。两相电动机（见图 9-25），常常用于轻型任务的应用中，三相电动机常常用在可变磁阻步进电动机中（见图 9-24），四相电动机用于需要更高功率的场合。

2. 步进角

对应定子线圈一个脉冲信号的变化时，转子转过的角度。

3. 保持转矩

通电时，使电机维持于静止位置且不引起主轴旋转的最大转矩。

4. 引入转矩

引入转矩是指步进电动机能够与输入信号同步启动，而不失步的最大转矩。

5. 失步转矩

失步转矩（脱出转矩）是指步进电动机能够在给定的步进速率下运转，而不失步的最大转矩。

6. 引入频率

步进电动机可以启动规定的负载而不失步的最高脉冲切换频率。

7. **失步频率**

失步频率是指在频率减小的情况下，负载电动机仍能保持同步的频率。

8. **扭转区域**

扭转区域是介于引入频率与失步频率之间的脉冲频率范围，在此频率范围内电动机工作同步，但不能启动或反转。

图 9-27 给出了一台步进电动机的一般特性曲线。

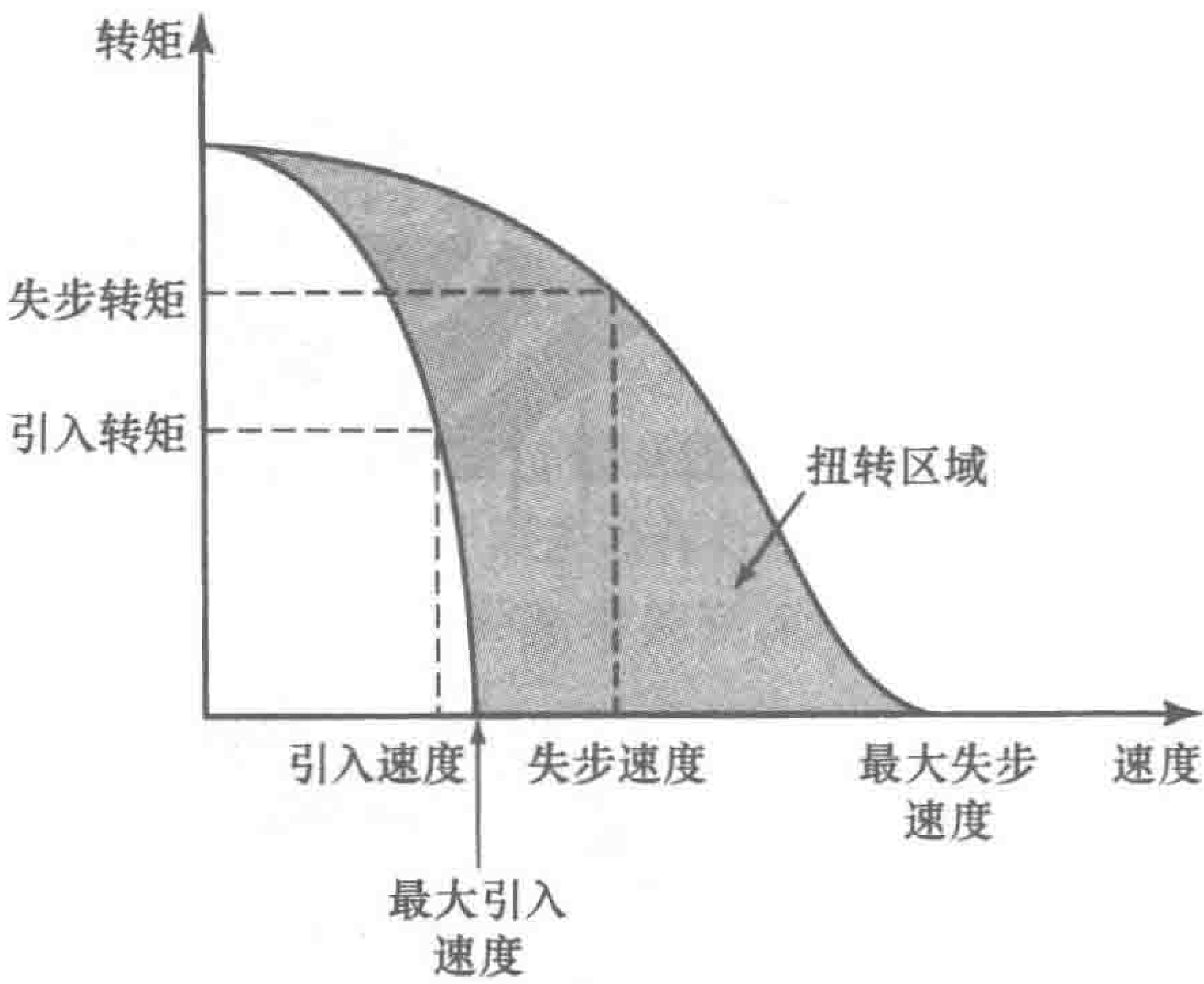


图 9-27　步进电动机的特性

9.7.2　步进电动机控制

固态电子技术用来在定子线圈之间切换直流供电。两相电动机，如图 9-25 所示，当它们有 4 根连接信号的导线生成切换序列时称为双极电动机（见图 9-28a）。这样的电动机可以用 H 电路驱动（见图 9-19 和相关讨论）；图 9-28b 给出了电路，表 9-2 给出了晶体管的切换顺序，共执行 4 步。然后 229 这个序列被重复循环，以执行更多的步数。这个序列给出了顺时针的旋转。如果要实现逆时针旋转，只需将序列颠倒。

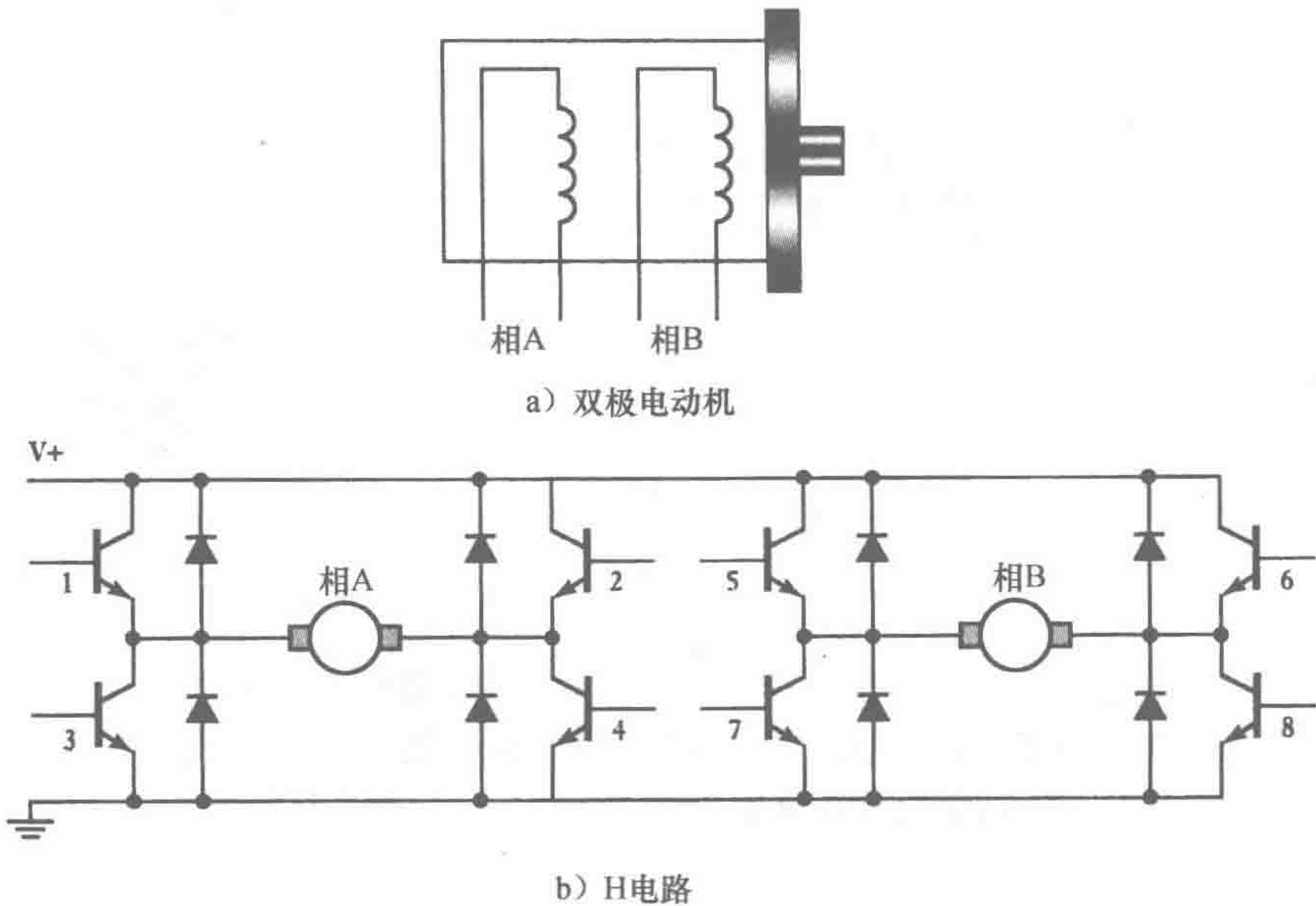


图 9-28　双极电动机

表 9-2　整步双极步进电动机的开关顺序

步	晶体管			
	1 和 4	2 和 3	5 和 8	6 和 7
1	开	关	开	关
2	开	关	关	开
3	关	开	关	开
4	关	开	开	关

如果不用整步序列来实现磁极从一步到另一步的旋转，而用半步序列切换线圈使转子停在一个半步的位置上，那么可以得到更精细的分辨率。表 9-3 给出了双极步进电动机的半步序列。

表 9-3　半步双极步进电动机的开关顺序

步	晶体管			
	1 和 4	2 和 3	5 和 8	6 和 7
1	开	关	开	关
2	开	关	关	关
3	开	关	关	开
4	关	关	关	开
5	关	开	关	开
6	关	开	关	关
7	关	开	开	关
8	关	关	开	关

当两相电动机有 6 根用于生成切换序列的连接线时（见图 9-29），称为**单极**步进电动机。每一根线圈都有一个中心抽头。当相线圈的中心抽头连在一起时，这种类型的步进电动机可以仅使用 4 个晶体管来实现开关。表 9-4 给出了用于产生顺时针旋转的晶体管的开关序列，然后这个序列重复循环，以执行更多的步数。对于逆时针旋转，只需将该顺序颠倒。表 9-5 给出了单极电动机工作在半步模式下的开关顺序。

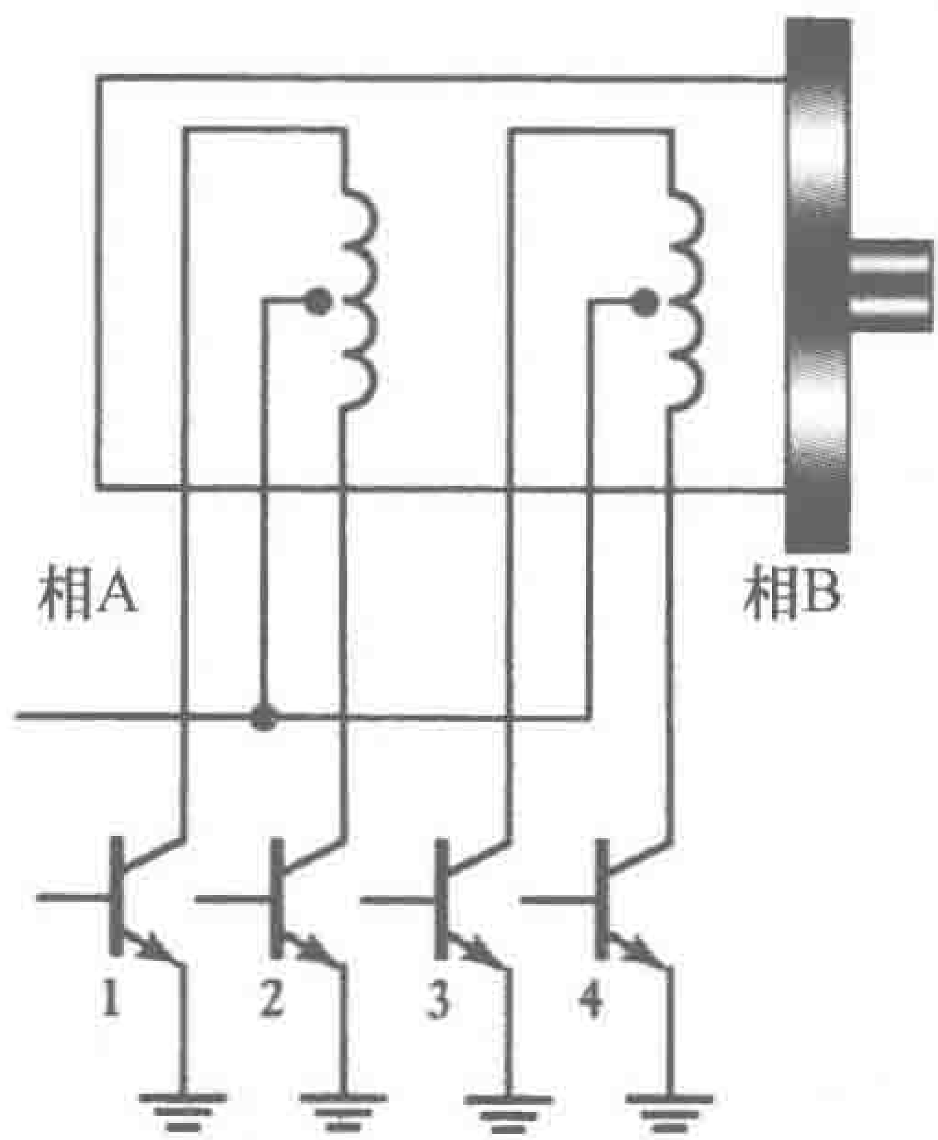


图 9-29　单极电动机

表 9-4　整步单极步进电动机的开关顺序

步	晶体管			
	1	2	3	4
1	开	关	开	关
2	开	关	关	开
3	关	开	关	开
4	关	开	开	关

表 9-5　半步单极步进电动机的开关顺序

步	晶体管			
	1	2	3	4
1	开	关	开	关
2	开	关	关	关
3	开	关	关	开
4	关	关	关	开
5	关	开	关	开
6	关	开	关	关
7	关	开	开	关
8	关	关	开	关

集成电路能用来提供驱动电路。图 9-30 给出了四相步进电动机集成电路 SAA 1027 的连接。通过加载高或低的信号来控制三个输入端。当置位端处于高电平时，集成电路的输出随着触发端电平从低到高变化一次，集成电路的输出状态就改变一次。序列每四步间隔重复一次，但通过在触发端加载低电平信号可复位为零状态。当旋转输入端是低电平时，顺时针旋转；高电平时，逆时针旋转。

230
≀
231

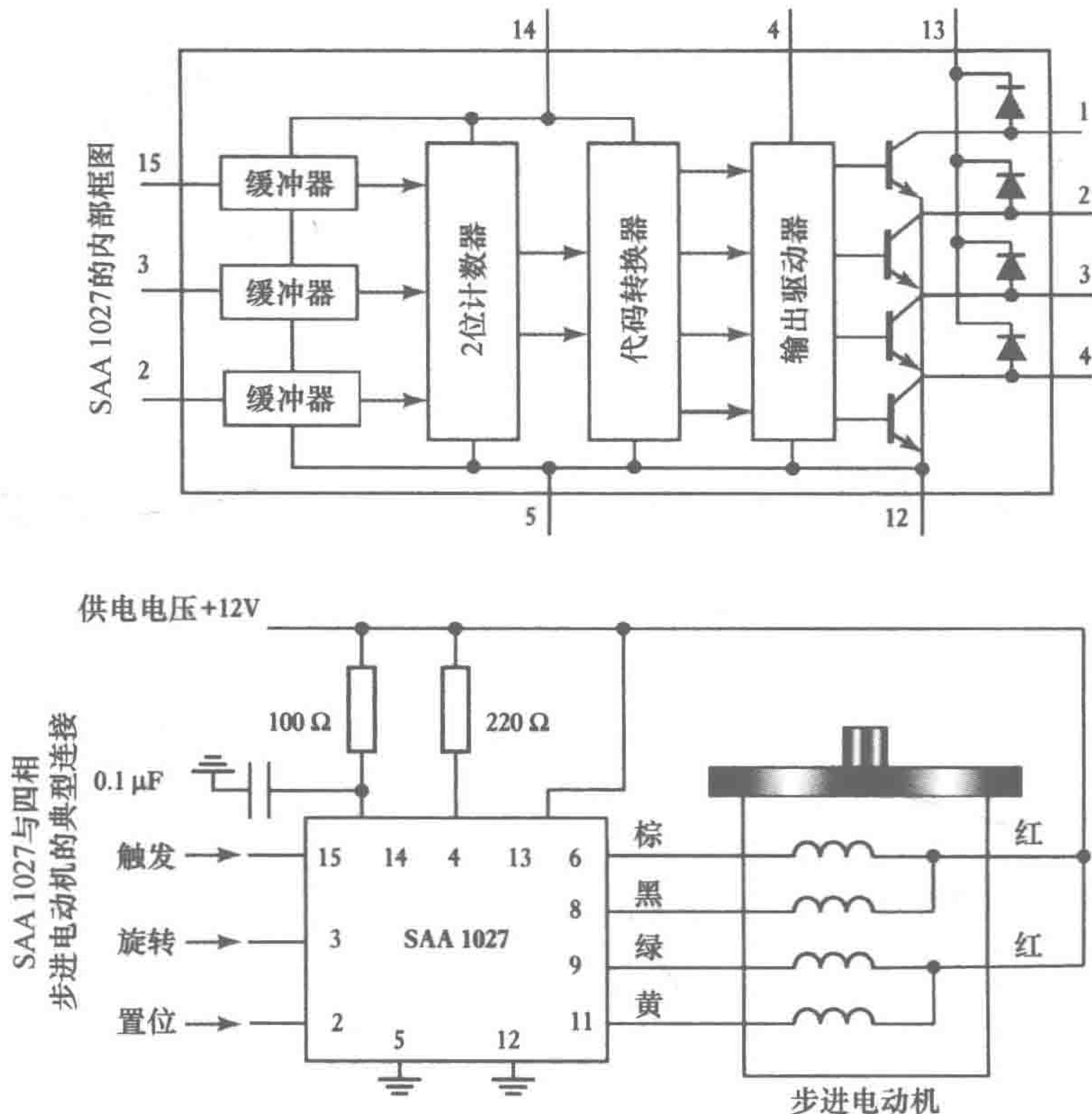


图 9-30　用于步进电动机的集成电路 SAA 1027

有些应用需要非常小的步进角。尽管步进角可以通过增加转子齿数或相位数来减小，但当超过四相和齿数在 50～100 时就不采用了。取而代之的是一项称为**微步进**的技术。该技术把每一步再细分为一些大小相等的步。这通过向线圈提供不同的电流来使转子转到两个正常位置的中间位置。例如，一个 1.8°的步进可再分为 10 个相等的步进。

步进电动机不但可以用来提供可控的旋转步进，也可用来提供连续旋转，并通过控制加载到它上面的脉冲的频率来控制旋转速度。这提供了一个非常有用的可控变速电动机，它有很多实际应用。

因为步进线圈有电感并且关闭电感负载能引发很大的反向电动势，所以当步进电动机连接到微处理器的输出接口时，有必要增加保护以避免对微处理器造成损害。这可以通过增加电阻来限制电流，尽管这可能需要仔细挑选电阻值使它既能够保护微处理器，又能够使电流不低于能接通晶体管的值。线圈中的二极管防止了电流反向流动，起到了保护作用。另一个可选择的方案是用光隔离器（见 3.3 节）。

232

9.7.3　步进电动机的选择

选择步进电动机应该考虑以下几点：

1）应用的工作转矩需求。额定转矩必须足够大，以满足转矩和扭转区域的需求。另外转矩 - 速度特性必须合适。

2）步进角必须有足够高的分辨率来提供需要的输出运动增量。

3）成本。

这将需要查看步进电动机的数据参数。以下的是从制造商数据表摘录的一台单极步进电动

机的典型数据（Canon 42M048C1U-N）：

直流工作电压	5V
每个线圈的电阻	9.1Ω
每个线圈的电感	8.1mH
保持转矩	66.2mNm/9.4 oz. in
转子转动惯量	$12.5\times10^{-4}gm^2$
定位转矩	12.7mNm/1.8oz. in
步进角	7.5°
步进角公差	±0.5°
每转步数	48

定位转矩（detent torque）是指步进电动机在没有通电的情况下，定子锁住转子的转矩。

电动机选好以后，需要一个与电机兼容的驱动系统。举例来说，对于一台单极电动机（最大输入电压为7V，最大单相电流为80mA），CyberneticsCY512 可能用到。对于小的单极步进电动机（最大输入电压18V，最大单相电流350mA），SigneticsSAA1027 驱动器广泛应用。对于一个两相双极或四相单极电动机来说，可以考虑使用 SCS - Thomson L297/L298 这个双芯片逻辑驱动器。L297 芯片生成了针对两相和四相单极电动机的四相 TTL 逻辑信号的电动机相位序列。L298 是一个桥式驱动器，功能为接收这样的信号并驱动电感负载，在本例中是步进电动机。一台双极电动机可以用2A 以上的绕组电流驱动。

当脉冲加载到步进电动机时，本质上是对电感 - 电阻电路加载了一个输入，产生的转矩加载到负载上，从而产生角加速度。由于系统有一个自然频率；所以电动机不会直接到达下一个步进位置，通常在到达稳态值之前，在下一个步进位置附近有阻尼振荡（见图 9-31）。详见 24.1.2 节的讨论和自然频率、阻尼系数的推导。

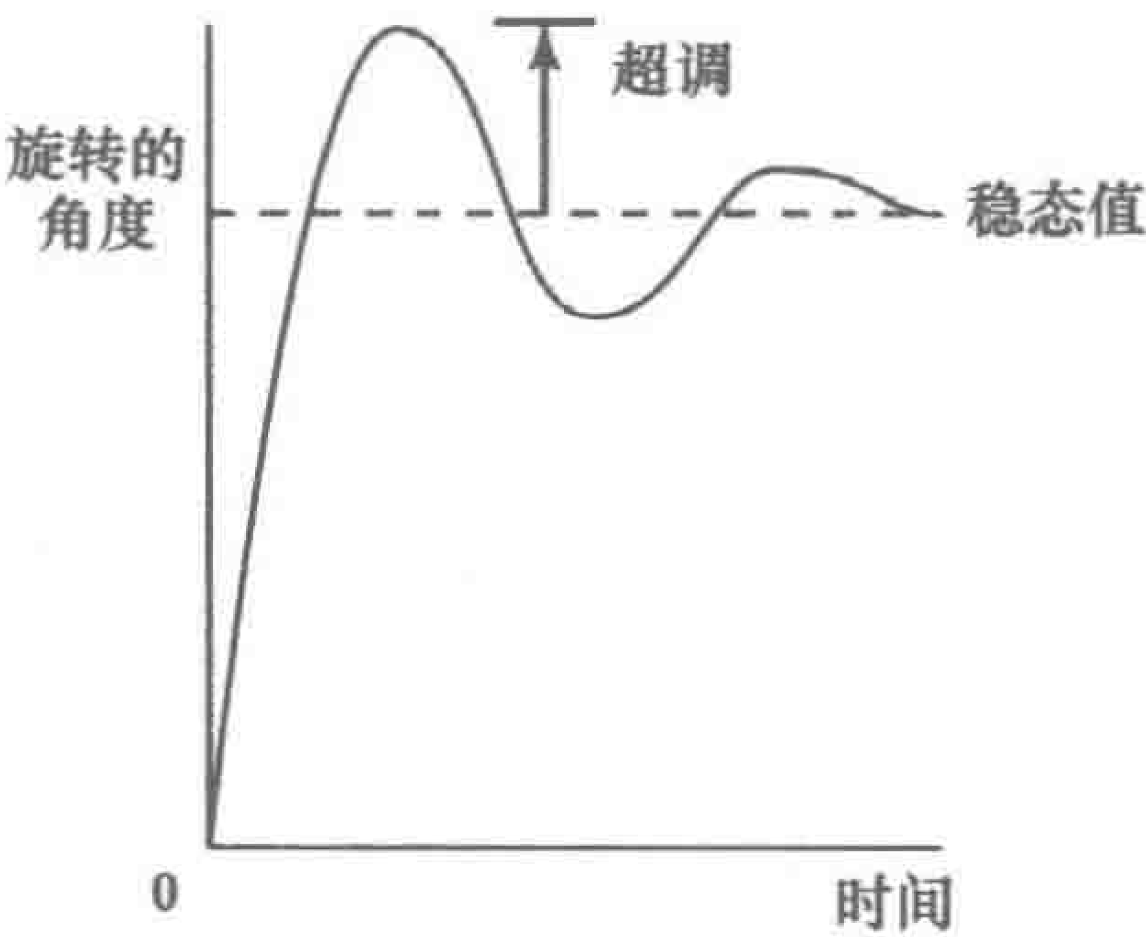


图 9-31　在稳态角度附近的振荡

233

9.8　电动机选择

当针对某种特定应用选择电动机时，需要考虑的因素有：

1）惯量匹配。

2）转矩要求。

3）功率要求。

9.8.1　惯量匹配

3.8 节中介绍的电抗的阻抗匹配可以引申到机械系统中，与电路类似的情况是电动机、转矩源和直接旋转的负载（见图 9-32a）。若负载的转动惯量为 I_L，产生的角加速度为 α，则所需的转矩是 $I_L\alpha$。加速电动机轴所需的转矩是 $T_M=I_M\alpha_M$，加速负载所需的转矩是 $T_L=I_L\alpha_L$。在没有齿轮装置的情况下，电动机轴与负载有相同的角加速度和角速度。加速系统所需的总功率是 $T_M\omega+T_L\omega$，ω 是角速度。因此：

$$功率 = (I_M + I_L)\alpha\omega$$

功率是由电动机转矩 T_M 产生的，因此功率应该等于 $T_M\omega$。因此：

$$T = (I_M + I_L)\alpha$$

234

当 $I_M=I_L$ 时，获得一个角加速度的转矩将达到最小。因此，为了传递最大功率，负载的转动惯量应该和电动机的转动惯量相似。

对于这种情况，当电动机通过齿轮传动转动负载时（见图9-32b），最大功率传递的条件是电动机的转动惯量等于负载折算到电动机轴上的转动惯量，即 $n^2 I_L$，这里 n 是齿数比，I_L 是负载的转动惯量（见17.2.2节）。

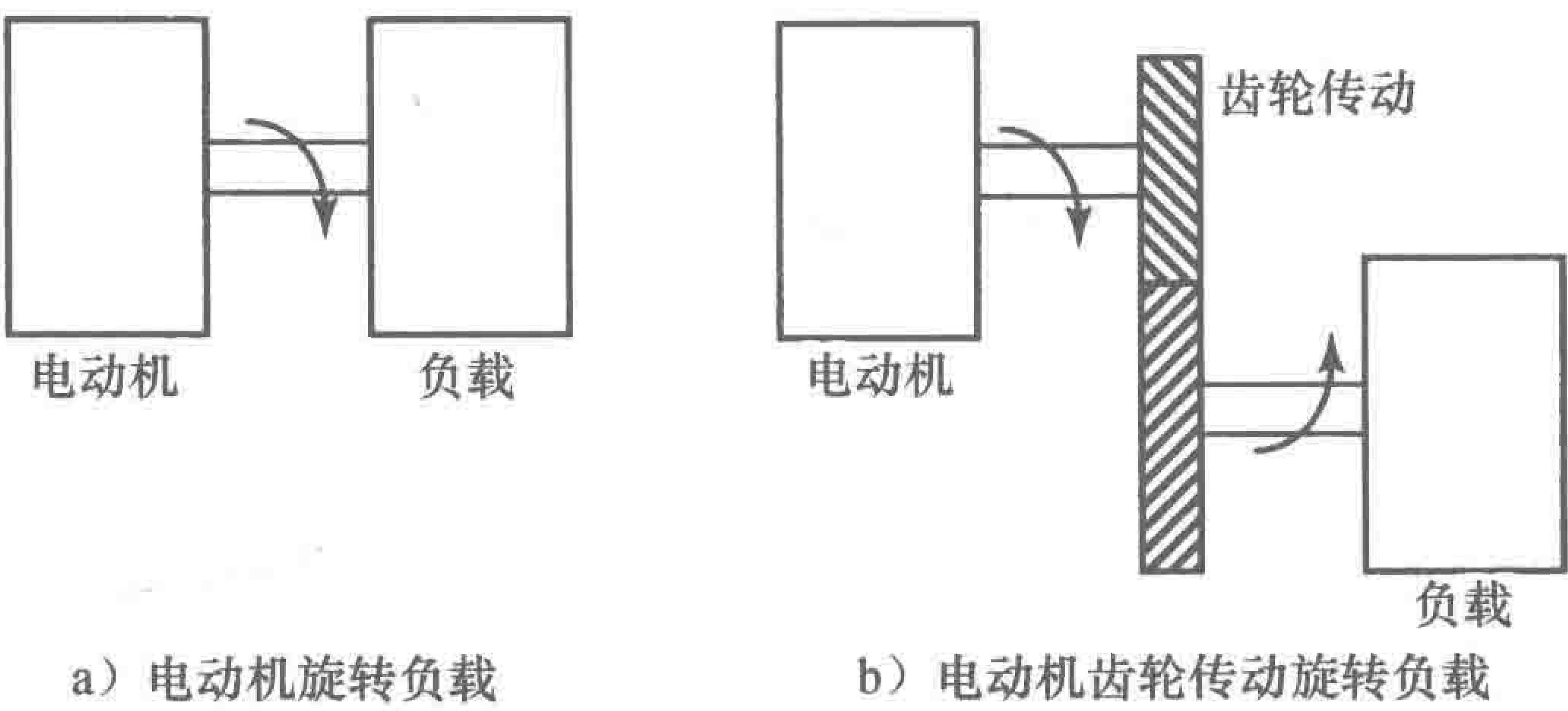


图9-32　电动机与负载简图

因此对一个最大功率传递，电动机的转动惯量应该等于负载的转动惯量。在有齿轮传动时，等于负载折算到电动机轴上的转动惯量。这意味着获得一个给定加速度的转矩将达到最小。如果电动机用在快速定位时这将特别有用。在有齿轮的系统中，可以调整齿轮传动比来实现转动惯量的匹配。

9.8.2　转矩要求

图9-33给出了一个典型电动机的工作曲线。对于连续运行，不应超过堵转转矩。这是保证在电动机不会过热的情况下，可以采用的最大转矩。对于间断使用，可以采用更大的转矩。随着角速度的增大，电动机传递转矩的能力降低。因此如果需要更高的速度和转矩，就需要选择一台功率更大的电动机。

假设我们需要一台电动机去操作一台滚筒式起重机升起一个重物（见图9-34）。如果滚筒直径为0.5m，最大载荷 $m=1000\text{kg}$，那么缆绳中的张力 $mg=1000\times9.81=9810\text{N}$。滚筒上的转矩为 $9810\times0.25=2452.5\text{Nm}\approx2.5\text{KNm}$。如果起重机以恒定速度 $v=0.5\text{m/s}$ 工作，因此滚筒的角速度为 $\omega=v/r=0.5/0.25=2\text{rad/s}$，或 $2/2\pi=0.32\text{revs/s}$。电动机通过齿轮驱动输出轴。在电动机最大速度约为1500rev/min或25rev/s时，我们可确定齿轮传动比为 $n=25/0.32$ 或接近80/1。电动机上负载引起的转矩将为滚筒上的转矩除以80，约为2500/80 = 31.25Nm。如果考虑齿轮的摩擦，电动机允许的最大转矩是35Nm。

235

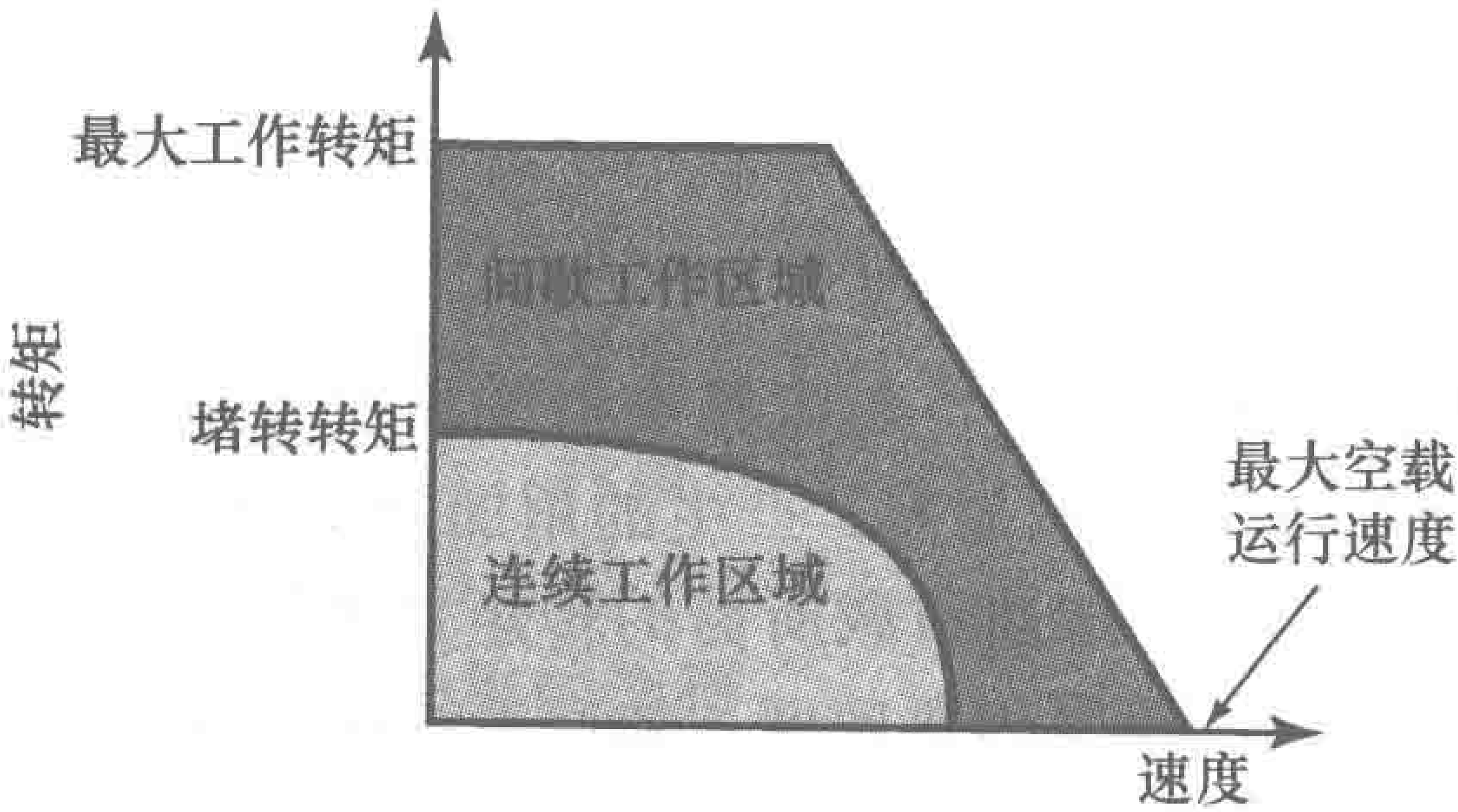


图9-33　转矩－速度图

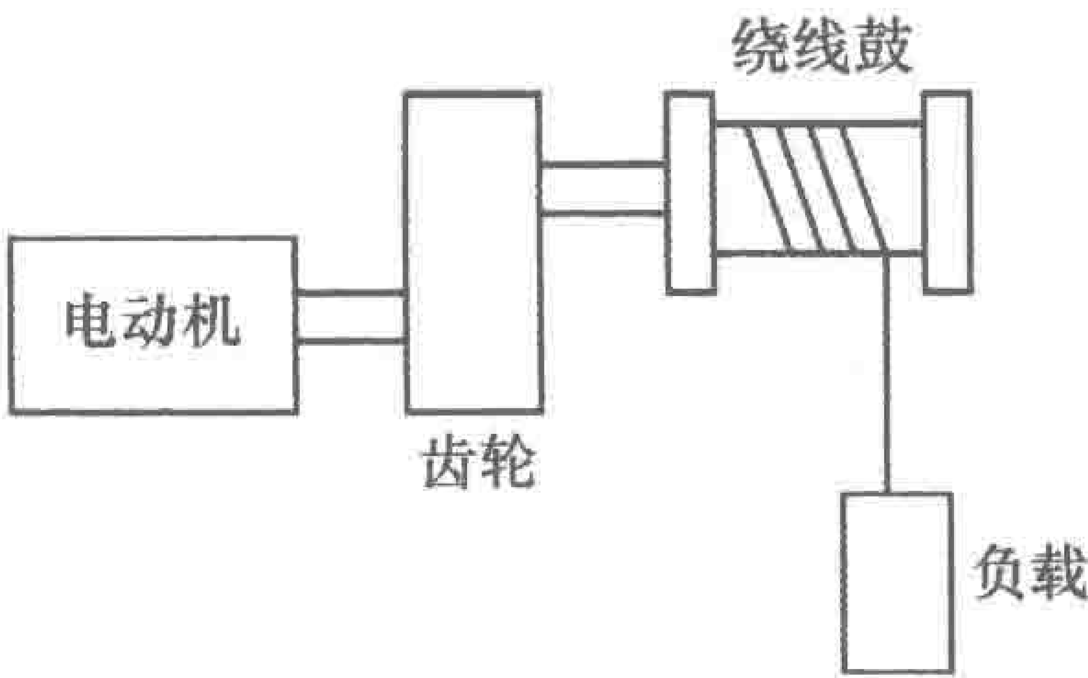


图9-34　一台提起负载的电动机

然而，这仅是当负载在恒定速度下往上提时的最大转矩。我们需要考虑从静止加速到0.5m/s的加速度所产生的转矩。假如说我们希望在1s内从静止达到这个速度，需要的加速转矩为 $I\alpha$，这里 I 是转动惯量，α 是角加速度。电动机通过传动装置给负载的有效转动惯量为 $(1/n^2)\times$ 负载转动惯量 mr^2，代入数值为 $(1/80^2)\times1000\times0.25^2=0.0098\text{kgm}^2$ 或约 0.01kgm^2。

滚筒和齿轮会增加0.02kgm^2 的转动惯量。为了计算升降装置的总转动惯量，我们必须再加上电动机的转动惯量。制造商的数据手册可能会给出一些值，比如0.02kgm^2，因此包含升降负载的总转动惯量为0.01+0.02+0.02=0.05kgm^2。电动机速度需要在1s内从0升到25rev/s，因此角加速度为（25×2π）/1=157rad/s^2，大约160rad/s^2。因此需要的加速转矩为0.05×160=8Nm。我们考虑的最大转矩是在恒定速度下提起重物的转矩，加上加速重物从0到此速度所需的转矩，大约为35+8=43Nm。

我们可用代数方法表述上例。从电动机处获得的转矩 T_m 是负载转矩 T_L 所需要的（如果有传动装置，齿轮传动比为 n，则为 T_L/n），再加上需要加速电动机的转矩 $I_m\alpha_m$，其中 I_m 是电动机转动惯量，α_m 是它的角加速度：

$$T_m = \frac{T_L}{n} + I_m\alpha_m$$

负载角加速度 α_L 由下式给出：

$$\alpha_m = n\alpha_L$$

因为有转矩 T_f 来克服负载摩擦，所以加速负载的转矩为（$T_L - T_f$），因此：

$$T_L - T_f = I_L\alpha_L$$

因此我们可以写成：

$$T_m = \frac{1}{n}[T_f + \alpha_L(I_L + n^2 I_m)]$$

9.8.3 功率要求

电动机需要在不过热情况下以需要的最大速度工作。总功率 P 包含克服摩擦的功率和加速负载的功率。由于功率是转矩和角速度的乘积，所以克服摩擦转矩 T_f 需要的功率为 $T_f\omega$。加速负载使其角加速度为α，需要的功率为（$I_L\alpha$）ω，这里 I_L 是负载转动惯量。因此：

236

$$P = T_f\omega + (I_L\alpha)\omega$$

本章小结

继电器是电控开关，通过改变一个电路的电流，打开或关闭另一个电路的电流。

二极管可以认为只允许电流朝一个方向流动，在相反方向电阻会很大。

晶闸管可以认为是特殊的二极管，有一个可以控制二极管导通条件的门极（控制级）。一个三端双向可控硅开关元件与晶闸管相似，并且等价于两个晶闸管在同一个芯片上反向并联。

通过把基极电流在0和使晶体管达到饱和的电流之间切换，**双极型晶体管**可用做开关。**MOSFET**和它类似，也可用做开关。

直流电动机的基本原理是：当在环形电枢线圈中通过电流时，电枢可以在磁场中自由转动。磁场可由永磁体或电磁体（即励磁线圈）产生。永磁电动机的速度由通过电枢线圈的电流决定；有励磁线圈的电动机的速度由电枢线圈或励磁线圈上的电流决定。这样的直流电动机需要换向器和电刷来周期性地转换每一个电枢线圈中的电流方向。**无刷永磁直流电动机**有一个永磁体转子和一系列定子线圈，这些定子线圈中的电流按顺序切换。

交流电动机可分为单相和多相两大类，每一类还可以细分为感应电动机和同步电动机。单相电动机常用于低功率的场合，而多相电动机常用于高功率需求的场合。感应电动机比同步电动机便宜，因此得到广泛使用。

步进电动机是一种电动机，当数字脉冲加载到输入端时这种电动机等角度转动，即所谓的步进。

电动机选择需要考虑**惯量匹配**、**转矩要求**和**功率要求**。

习题

9.1 解释图 9-35 中的电路如何防止开关的弹跳?

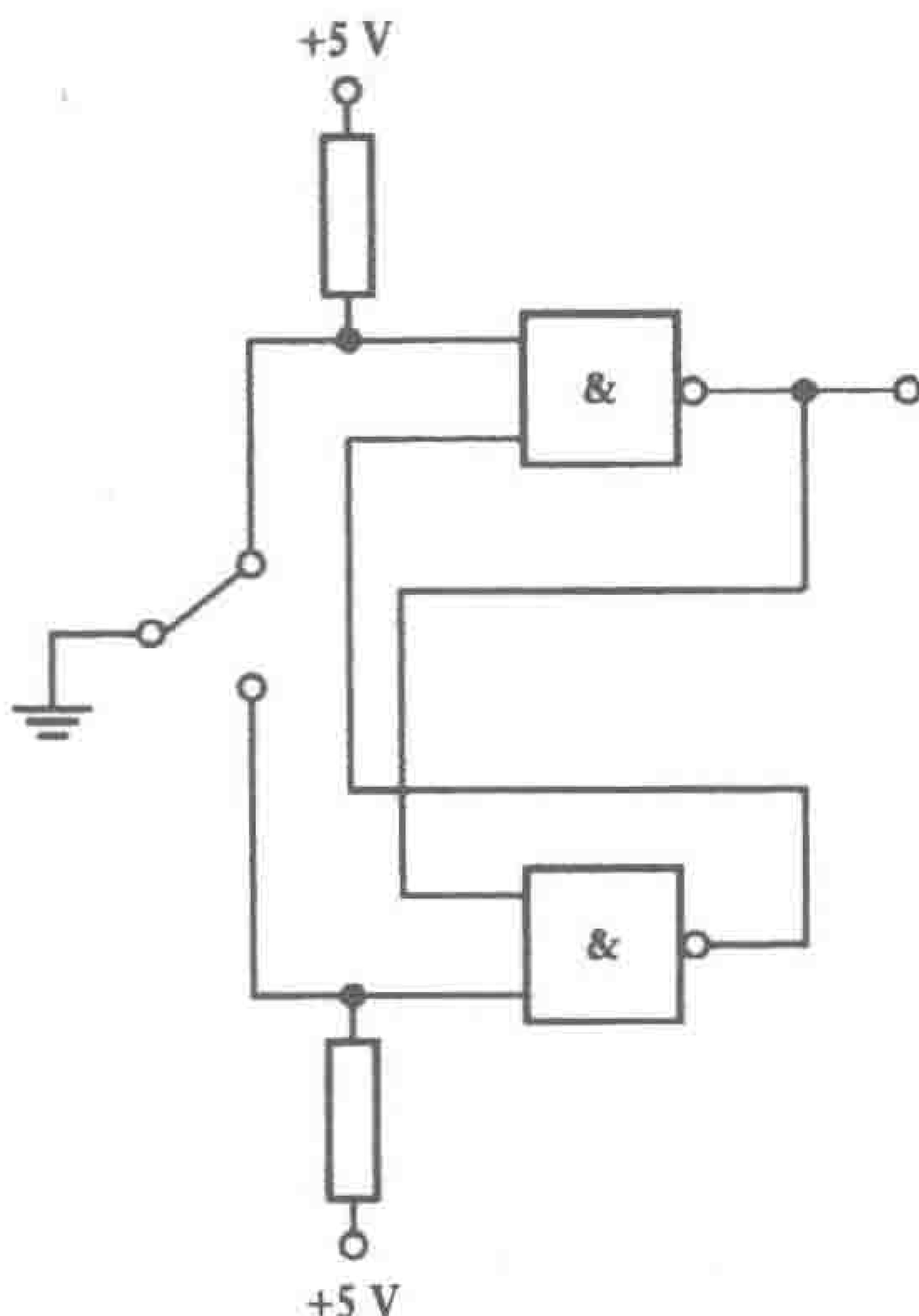


图 9-35 习题 9.1

9.2 解释晶闸管如何通过对一个不变的电源电压输出进行斩波来控制直流电压的电平?

9.3 一台直流电动机需要具备(a)重载低速时的高转矩;(b)无论速度多少,都有较稳定的转矩。找出合适的电动机类型。

237

9.4 针对以下条件找出可能的直流或交流电动机。(a)便宜的,需要稳定的转矩操作;(b)需要可控制的高速;(c)需要低速;(d)维护需求最小化。

9.5 解释无刷直流永磁电动机的原理。

9.6 解释可变磁阻步进电动机的操作原理。

9.7 如果一台步进电动机的步距角为 7.5°,要产生 10rev/s 转速,需要多大的数字输入频率?

9.8 一台带有 8 个定子绕组和 10 个转子齿数的混合步进电动机需要多大的步距角?

9.9 一台永磁直流电动机电枢电阻为 0.5Ω,当加载 120V 电压时,此电动机达到稳定状态转速为 20rev/s,电流 40A。求:(a)电动机的输入功率;(b)电枢中损失的功率;(c)此速度下产生的转矩?

9.10 如果一台直流电动机在电枢电流为 2A 时产生了 2.6Nm 的转矩,当电流为 0.5A 时转矩是多少?

9.11 如果一台电动机需要输出转速 0.25rev/s 和步距角 7.5°,微处理器每秒需要产生多少步进或脉冲输入到这台步进电动机?

9.12 一台步进电动机用来转动直径为 240mm 的滑轮,皮带上运载物重 200kg。如果此物体 2s 内从静止匀速加速到 100mm/s,并且存在一个 20N 的恒定摩擦力,则电动机需要的引入转矩(pull-in torque)是多大?

238

第四部分　微处理器系统

第 10 章　微处理器和微控制器

本章目的

通过本章的学习，读者将能够：

- 描述微处理器系统的基本结构。
- 描述常见微处理器的架构，以及它们是如何被集成到微处理器系统中的。
- 描述微控制器的基本结构，以及如何设置其寄存器来执行任务。
- 解释如何用流程图和伪代码描述程序。

10.1　控制

举一个简单的控制问题，比如十字路口红、黄、绿交通灯的时序控制，由组合和时序逻辑电路组成的电子控制系统是能够胜任的。但是，在更复杂的情况下可能会出现以更复杂的时序来控制更多变量的情形。当今最简单的解决方法已经不是建立一个基于硬件连接的组合和时序逻辑电路的系统，而是利用微处理器和软件来实现“互连”。

本书所涉及的微处理器系统是特指用于控制的系统，称为**嵌入式微处理器**。这是因为这种微处理器主要用于实现某种特定的功能。同时，它是自启动的，即不需要人为干预，并且能完全独立地执行自身的程序。在一般人看来，该系统作为一个微处理器系统的特征并不明显。因此，虽然现代洗衣机配备了微处理器，但操作者必须通过按下合适的按钮或旋转一个开关，然后点击开始按钮来选择洗衣模式。

本章是微处理器和微控制器结构的概述，而接下来两章将讨论编程。第 13 章将介绍接口技术。

10.2　微处理器系统

239
~
241

应用微处理器的系统有三个基本部分：**中央处理器**（CPU），用于识别和执行程序指令（这是使用微处理器的部分）；**输入/输出（I/O）接口**，用于外界和微处理器直接的通信（术语**端口**用于接口）；**存储器**，用于存放程序指令和数据。图 10-1 为微处理器系统的通用结构框图。

在同一芯片上拥有存储器及多个输入/输出的微处理器称为**微控制器**。

10.2.1　总线

数字信号从一部分传输到另一部分的路径称为**总线**。从物理形式上来说，总线是多根并行的、可以用于信号传输的导线，并且可以被系统中的所有芯片共享。这是因为如果芯片之间都是分开连接的，那么将需要大量的导线。但是，使用芯片共用的总线意味着当某个芯片把数据送入总线时，其他芯片要等待直到数据传输完成后才能把数据送入总线。典型的总线有 16 或 32 根并行连接，其中每根导线均可以同时传输数据字的 1 位。这使得总线传输速度比串行传输数据快。串行传输是指一个完整的字是通过一根导线逐位发送的。

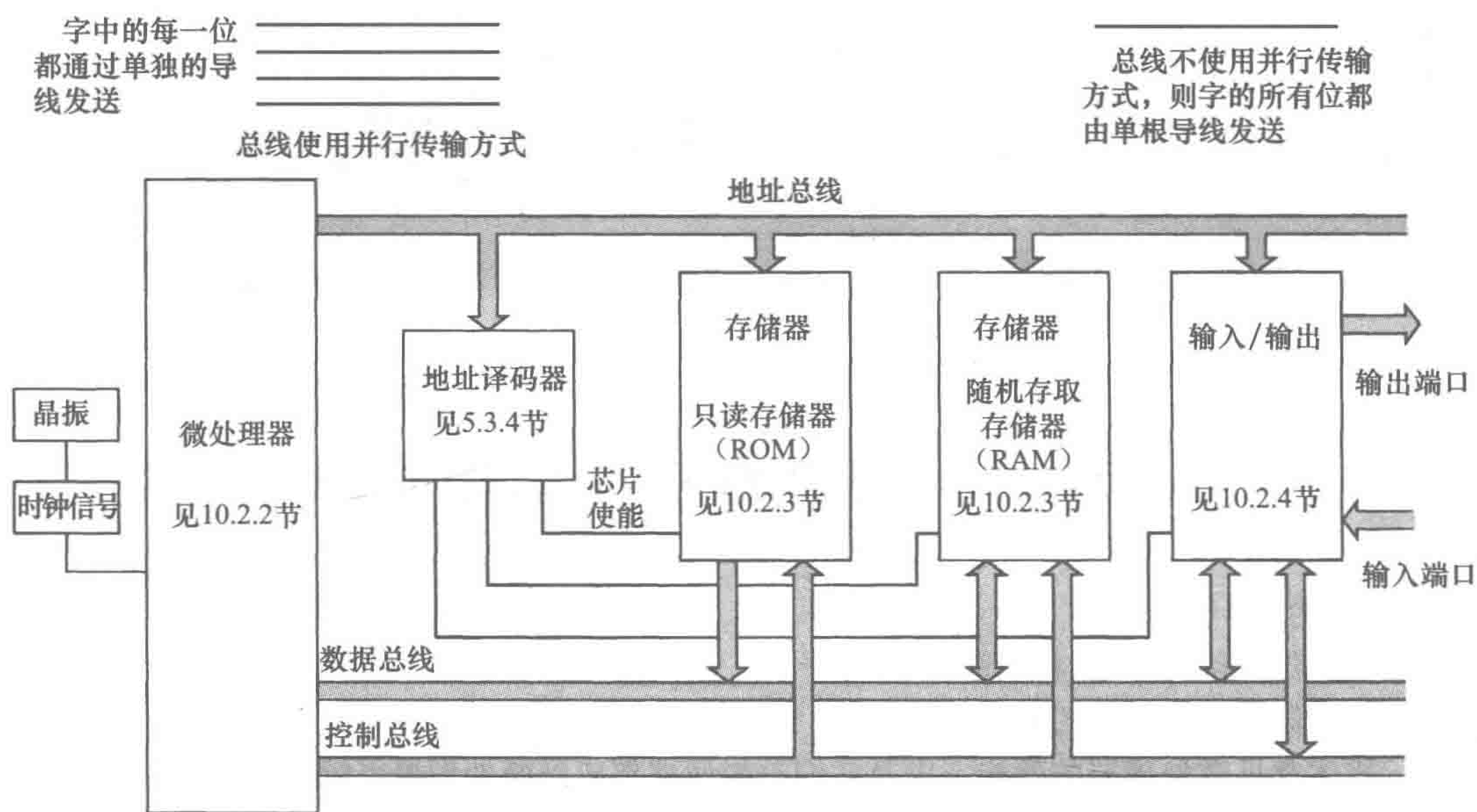


图 10-1 微处理器系统及其总线的一般形式。所有组件共享相同的数据总线和地址总线，这种结构被称为冯·诺依曼体系结构

在一个微处理器系统中有以下三种总线。

1. 数据总线

和中央处理器（CPU）处理功能有关的数据是通过**数据总线**传输的。因此，数据总线用于中央处理器和存储器或中央处理器和 I/O 口之间传输一个字。总线中的每根数据线均能传送 1 位信号，即 0 和 1。所以对于一个有 4 根线的总线，我们可以传送字 1010，其中每位都由总线中一根独立的数据线传送，如下表所示：

242

字	总线	字	总线
0（LSB）	第 1 根数据线	0	第 3 根数据线
1	第 2 根数据线	1（MSB）	第 4 根数据线

总线的数据线越多，能用的字长越长。一个单独的数据项取值范围受到字长所能表示的范围的限制。因此，对于一个 4 位的字，取值的数量为 $2^4=16$。如果这个数据要代表温度，则温度的可能范围必须要等分成 16 份。最早的微处理器是 4 位（字长）的设备，并且这种 4 位的微处理器至今仍在玩具、洗衣机和家用中央供暖控制器中广泛应用。随后发展为 8 位的微处理器，比如摩托罗拉 6800、Intel 8085A 和 Zilog Z80。现在还有 16 位、32 位和 64 位的微处理器，但是 8 位的微处理器仍然广泛应用。

2. 地址总线

地址总线传输的信号表示应该到哪里找到所需要的数据和特定存储器地址或 I/O 端口的选取。存储设备中的每个存储位置都有一个特定的编码，称为地址，因此系统能够在存储器中选取特定的指令或数据。每个 I/O 端口也同样有地址。当某个特定的地址被地址总线的地址信号所选取时，只有该地址所在的位置才与 CPU 通信。这样就能实现 CPU 每次只和一个位置通信。典型的 8 位数据总线计算机有 16 位宽的地址总线，即 16 根信号线。该地址总线能给 2^{16} 个位置进行编址；2^{16} 相当于 65 536 个位置，因此经常被写为 64K，其中 K 等于 1024。越多的存储空间能被编址，则更多的数据可被存储，就可以运行更庞大和复杂的程序。

3. 控制总线

有关控制行为的信号是通过**控制总线**传输的。例如，微处理器必须要通知存储设备是从输入装置读入数据还是把数据写入输出装置。“读取”是指接收信号，而“写入”是指发送信号。另外，控制总线也传输系统时钟信号来同步微控制器系统中的所有行动。该时钟是一个晶体控制振荡器，能够产生有固定时间间隔的脉冲。

10.2.2 微处理器

243

我们通常认为微处理器是中央处理单元，它是处理系统中负责处理数据、从存储器取指令、译码和执行的部分。不同的微处理器有不同的内部结构，即有不同的“架构”。图 10-2所示的是微处理器通用架构的简化图。

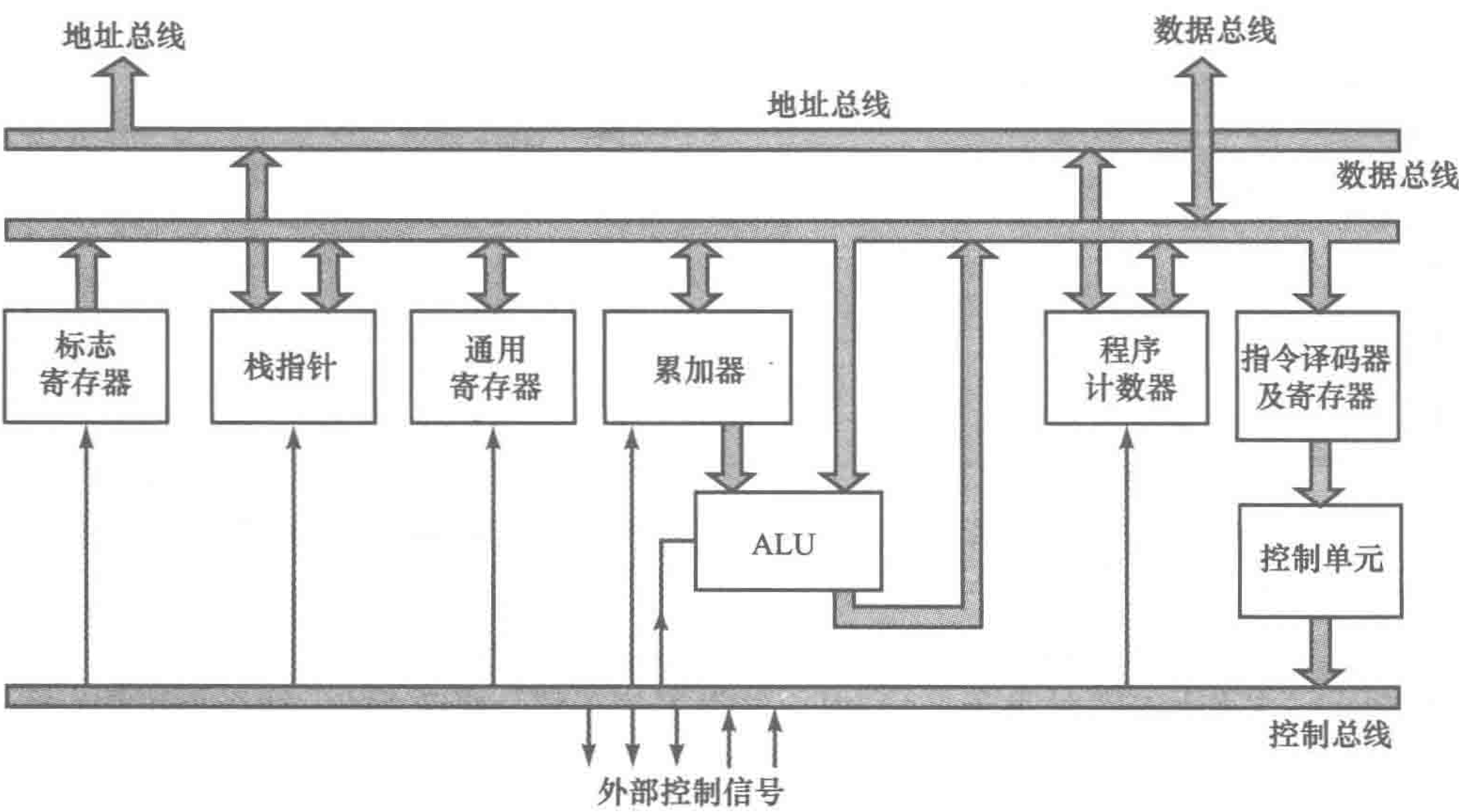


图 10-2 微处理器系统的通用内部架构

微处理器包括以下几个功能模块：

1. 算术逻辑单元

算术逻辑单元（ALU）负责完成数据操作。

2. 寄存器

在指令正在被执行时，CPU 正在使用的内部数据会暂时存放于一组寄存器中，该组位于微处理器的**寄存器**用于存放与程序执行有关的信息。一个微处理器应该有一组寄存器，其中每种不同类型的寄存器具有不同的功能。

3. 控制单元

控制单元决定了微处理器操作的时序。它产生用于从存储器获取和执行程序指令的时序信号。摩托罗拉 6800 使用的是最大频率为 1MHz 的时钟，也就是一个时钟周期为 1μs，而一个指令需要 2 ~ 12 个时钟周期。涉及微处理器操作的时间可以通过它所占用的周期数来推算。

寄存器可分为多种，而根据不同的微处理器，微处理器的数量、大小和类型均有所不同。以下是一些通用寄存器：

1. 累加寄存器

累加寄存器（A）是暂时存放算术和逻辑单元输入数据的地方。为了使 CPU 可以访问（即读）存储器中的数据和指令，必须通过地址总线提供的存储器地址。接下来所需要的指令或数据就能通过数据总线读入到 CPU 中。由于每次只能有一个存储位置被编址，所以当数字

244

要进行运算时，必须要使用暂存寄存器。例如，当两个数进行加法运算时，其中一个数取自某个地址并且被存放在累加器中，而 CPU 则从另外一个存储地址读取另外一个数，然后两者就可以通过 CPU 中的算术逻辑单元进行运算，最后把结果存放在累加器中。可见，累加器是算术逻辑单元处理数据的临时存放寄存器，同时也是操作完毕后存放结果的寄存器。因此，它涉及所有与算术逻辑运算有关的数据传输。

2. 状态寄存器或程序状态字寄存器

状态寄存器包含了算术和逻辑单元当前运算结果的有关信息。它的每一位都有特殊的意义，称为**标志位**。每个标志位均可被置位或复位来表示当前操作的状态。例如，它可以表明当前操作结果是否为负数、零、有进位（两个二进制数 1010 和 1100 相加后得到的结果（1）0110 比微处理器的字长要大，因此有进位 1）、溢出或者程序是否允许外部中断。以下是常见的标志位：

标志位	置位，即置 1	复位，即清 0
Z	结果是 0	结果是非 0
N	结果是负	结果是非负
C	产生进位	未产生进位
V	产生溢出	未产生溢出
I	忽略中断	中断被正常处理

作为一个例证，我们考察两个十六进制数 02 和 06 相加后的标志位 Z、N、C 和 V 的状态。相加后结果为 08，因为结果不等于零，则 Z 为 0。结果为正数，则 N 为 0。不存在进位，则 C 为 0。该无符号结果在 -128 ~ +127 之间，即无溢出，则 V 为 0。现在，我们考察两个十六进制数 F9 和 08，其相加后的结果为（1）01。结果不等于零，则 Z 为 0。结果为正数，则 N 为 0。无符号结果存在进位，则 C 为 1。该无符号结果在 -128 ~ +127 之间，则V 为 0。

3. 程序计数器（PC）寄存器或指令指针（IP）

该寄存器使 CPU 可以跟踪程序所在位置，其中存放的是下一条程序指令所在的存储空间的地址。当执行某条指令时，程序计数器将会更新，把下一条程序指令所在的存储空间地址存储起来，并且计数器累加，因此，除非有指令使计数器不顺序执行，比如 JUMP 或 BRANCH，否则 CPU 会按顺序执行指令。

245

4. 存储器地址寄存器（MAR）

该寄存器存放数据的地址。例如，当两个数相加时，第一个数对应的地址将被写入存储器地址寄存器，然后该地址对应的数据将传输到累加器中。接着第二个数对应的地址被写入存储器地址寄存器，然后该地址对应的数据和累加器中的数据相加。其结果将存放在存储器地址寄存器中的地址所对应的存储空间中。

5. 指令寄存器（IR）

该寄存器用于存放指令。CPU 从存储器经过数据总线获取指令后，将其存放在指令寄存器中。这样的取指操作结束后，微处理器使程序计数器加 1，使程序计数器指向下一条等待被获取的指令。这时 CPU 译码，并用于执行一个操作。该过程称为“取指 - 执行周期”。

6. 通用寄存器

该寄存器可用于暂时存放数据或地址，也可用于其他寄存器之间传输数据有关的操作中。

7. 栈指针寄存器（SP）

该寄存器中的内容定义了 RAM 中栈顶的地址。栈是存储器中的一块特殊区域，可在执行子程序时存放程序计数器的值。

寄存器的数量和类型取决于所涉及的微处理器。比如，摩托罗拉 6800 微处理器（见图 10-3）有两个累加器，一个状态寄存器，一个变址寄存器，一个栈指针寄存器和一个程序计数器。其状态寄存器有负数、零、进位、溢出、半进位和中断标志位。而摩托罗拉 6802 除了有类似的上述寄存器外，还包含一个容量更小的 RAM 和内置的时钟发生器。

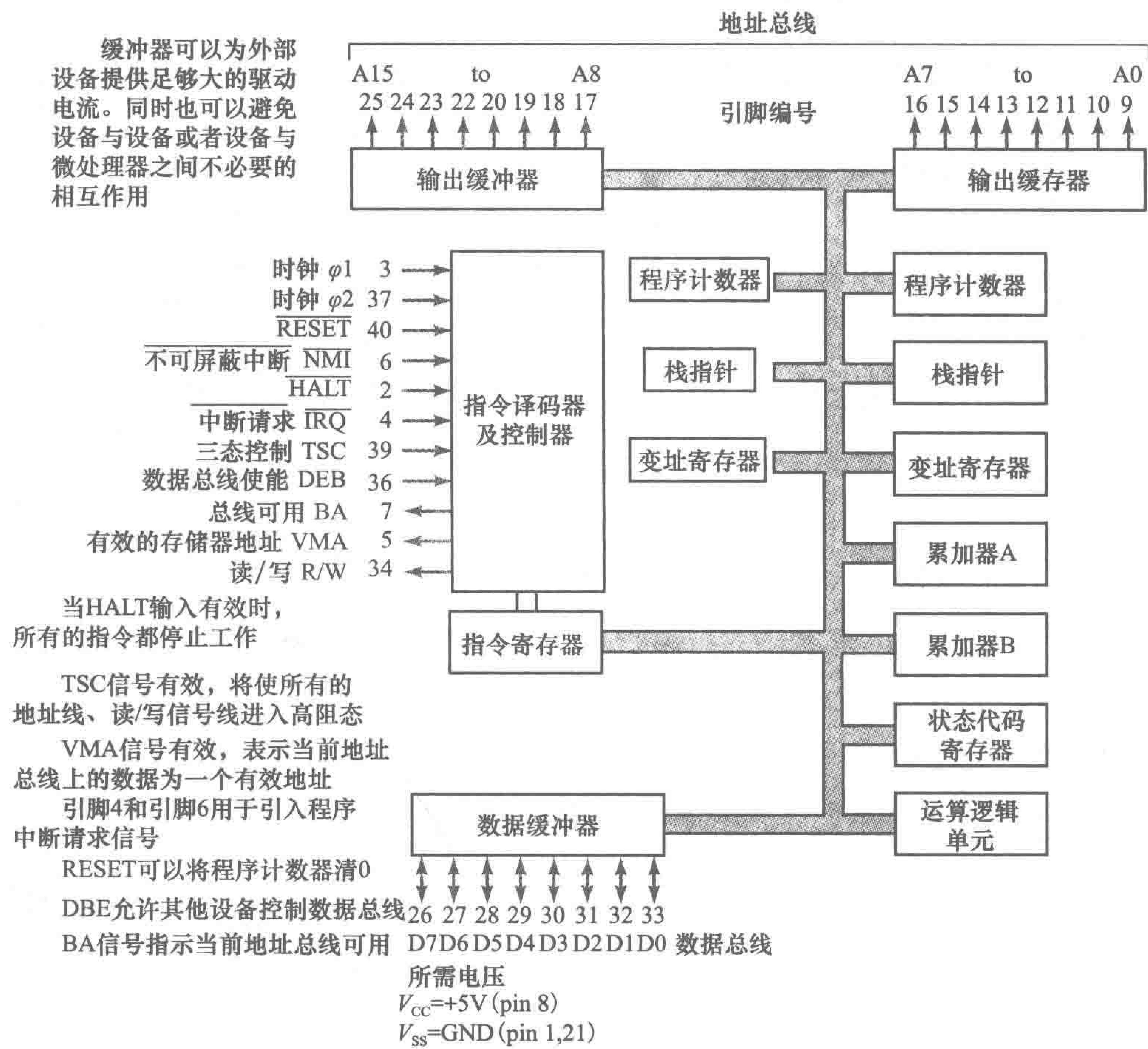


图 10-3 6800 系统结构

Intel 8085A 微处理器是早期 8080 微处理器的升级版。8080 版本需要外部时钟发生器，而 8085A 有内置的时钟发生器。针对 8080 所写的程序可以在 8085A 上运行。8085A 有 6 个通用寄存器，分别为 B、C、D、E、H 和 L，同时还有一个栈指针、一个程序计数器、一个标志位寄存器和两个暂存寄存器。其中 6 个通用寄存器可作为 8 位寄存器单独使用，也可作为 16 位寄存器成对使用，分别为 BC、DE 和 HL。图 10-4 为其结构的框图。

从图 10-3 和图 10-4 可明显看出，微处理器有一系列定时和控制的输入和输出。它们可以在微处理器正在执行某个操作时提供输出，也可提供输入来影响控制操作。另外，这些输入也与中断控制有关，它们允许程序的执行被外部事件中断。

10.2.3 存储器

微处理器系统中的存储器以一个或多个集成电路的形式存放二进制数据。这些数据可以是程序指令代码或者正在被操作的数字。

存储器的大小取决于地址总线中线的数量。存储元件是由大量能够存储 0 或 1 的存储单元

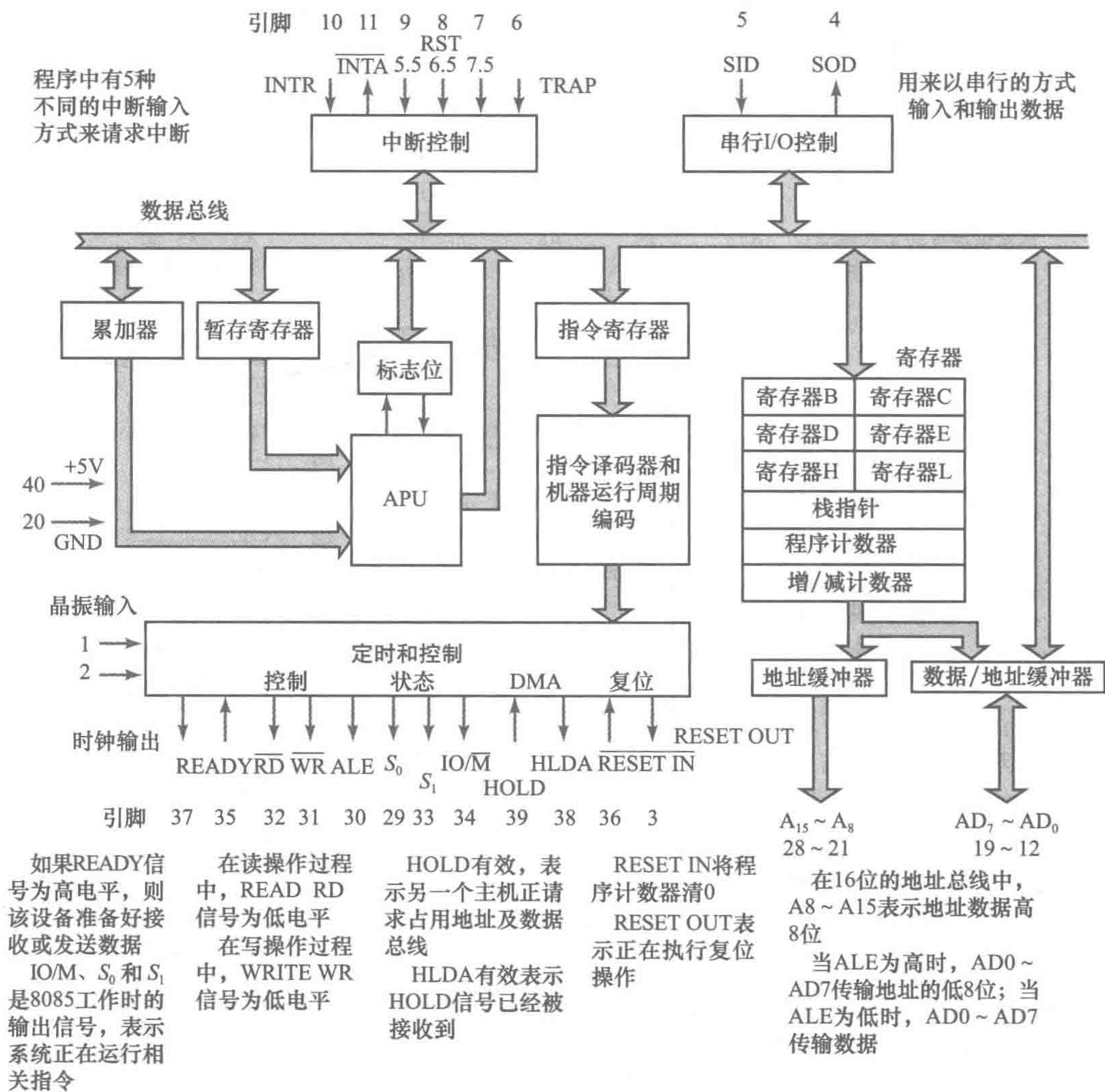


图 10-4　Intel 8085A 系统结构

组成的。这些存储单元按地址分组，每组存放一个字。为了访问这些存储字，每个位置均由独一无二的地址来确定。因此一个 4 位的地址总线可以有 16 个不同的地址，每个地址存储 1 字节，即 8 位一组的数据（见图 10-5）。

地址	数据内容
0000	
0001	
0010	
0011	
0100	
etc.	
1111	

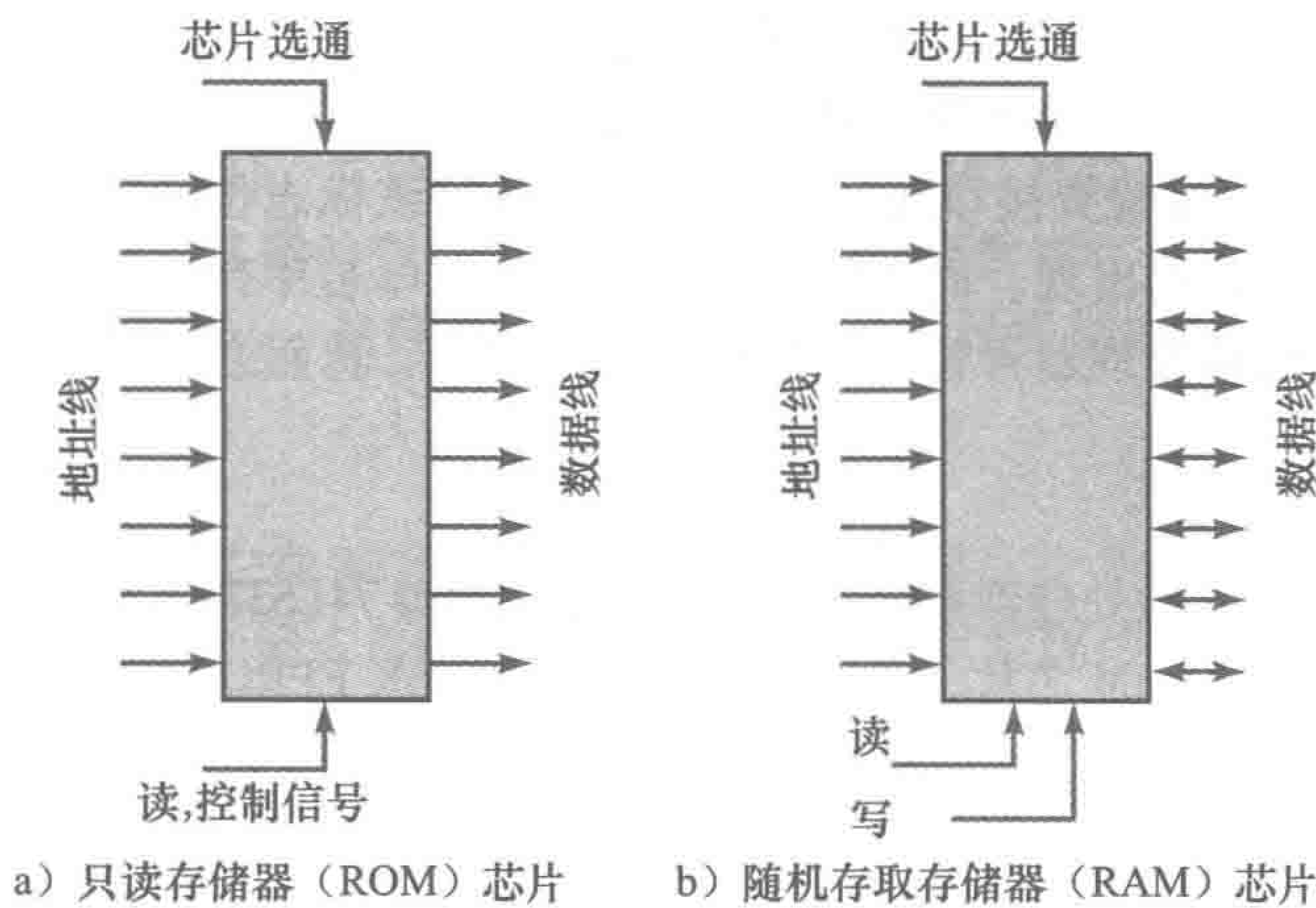
图 10-5　地址总线范围

存储器的容量定义为可用的存储地址的数量。1K 为 2^{10} = 1024 个存储地址，因此 4K 的存储器有 4096 个存储地址位置。

存储器分为以下几种：

1. 只读存储器

一个永久存储数据的存储器称为**只读存储器**（ROM）。ROM 内保存的内容在集成电路生产过程中已被编程烧录进去。当芯片在计算机中使用时，任何数据都不能写入该存储器中。ROM 中的数据只能被读取，且用于固定的程序，比如计算机操作系统和专门微处理器的应用程序。掉电时，ROM 中的数据不会丢失。图 10-6a 是典型只读存储器的引脚连线，它可存储 1K×8 位的数据。



a）只读存储器（ROM）芯片　b）随机存取存储器（RAM）芯片

图10-6　ROM和RAM

2. 可编程只读存储器

可编程只读存储器（PROM）是指用户可以对其编程的ROM。初始状态下，每个存储位都有一个可熔断的连接使其保持为0。通过给熔断丝输送电流使其永远断开，0将永久性地变成1。一旦可熔断的连接被打开，数据将永久地存储在存储器中且不能再修改。

3. 可擦写可编程只读存储器

可擦写可编程只读存储器（EPROM）指的是用户可以对其编程，且其内容可修改的ROM。一个典型的EPROM芯片包含一系列可以存储电荷的小型电路和单元。通过给集成电路引脚提供电压来产生一系列的注入电荷和释放电荷的单元，我们可以把程序存储起来。这一系列的单元将永久地存储在芯片中，除非向设备顶部的石英窗照射紫外线，这将导致所有的单元都放电。这时，就可以对该芯片重新编程。Intel 2716 EPROM有11根地址引脚和一个低电平有效的芯片使能引脚。

4. 电可擦可编程只读存储器

电可擦可编程只读存储器（EEPROM）类似于EPROM，不同的是它是通过加一个相对较高的电压来实现擦写而不是通过紫外线的照射。

5. 随机存取存储器

临时数据，即正在被操作的数据，存放在可读/写的存储器中，该存储器称为**随机存取存储器**（RAM）。可对该存储器读取，也可以写入。图10-6b是1K×8位RAM的典型引脚连接。摩托罗拉6810 RAM芯片有7根地址引脚和6根芯片选通引脚，其中4根选通引脚低电平有效，另外两根选通引脚高电平有效，并且只有当6根芯片选通引脚均有效时才能使RAM芯片有效。

当ROM被用做存储程序时，在系统启动时，程序是可访问的且已准备好被调用。存储在ROM中的程序称为固件。必须始终存在一些固件。当RAM用做存储程序时，该程序称为软件。当系统启动时，系统可能从外部设备（比如键盘、硬盘、软盘或者CD）把软件读入RAM中。

10.2.4 输入/输出

输入/输出操作定义为微处理器和外部世界之间的数据传输。**外部设备**是指用于和微处理器系统交换数据的装置。因为外部设备的速度和特性与微处理器明显不同，所以它们是通过接口芯片连接的。接口芯片的主要功能就是同步微处理器和外部设备之间的数据传输。在执行输入操作时，输入设备把数据放在接口芯片的数据寄存器中，该数据在被微处理器读取前一直存在。在执行输出操作时，微处理器同样地把数据放在寄存器中，直至该数据被外部设备读取。

246～249

在微处理器要从输入设备读取有效数据时，它必须确定接口芯片正确地锁存了输入数据。这可以通过查询或中断的方式来实现。对于查询方式，当接口芯片有有效数据时，它的一个标志位置1。微处理器不断地检查接口芯片，直到它检测到该标志位被置1。该方法的问题是微处理器必须等待该状态位的出现。而对于中断方法，当接口芯片有有效数据时，发送一个中断信号给微处理器；然后微处理器将暂停正在执行的主程序，转而去执行与该中断有关的程序来读取数据。

10.2.5　系统示例

图10-7是Intel 8085A微处理器系统。它包括地址锁存器74LS373、3-8地址译码器74LS138、两个型号为2114的1K×4RAM芯片、一个型号为2716的16K EPROM芯片以及输入/输出接口芯片74LS244和74LS374。

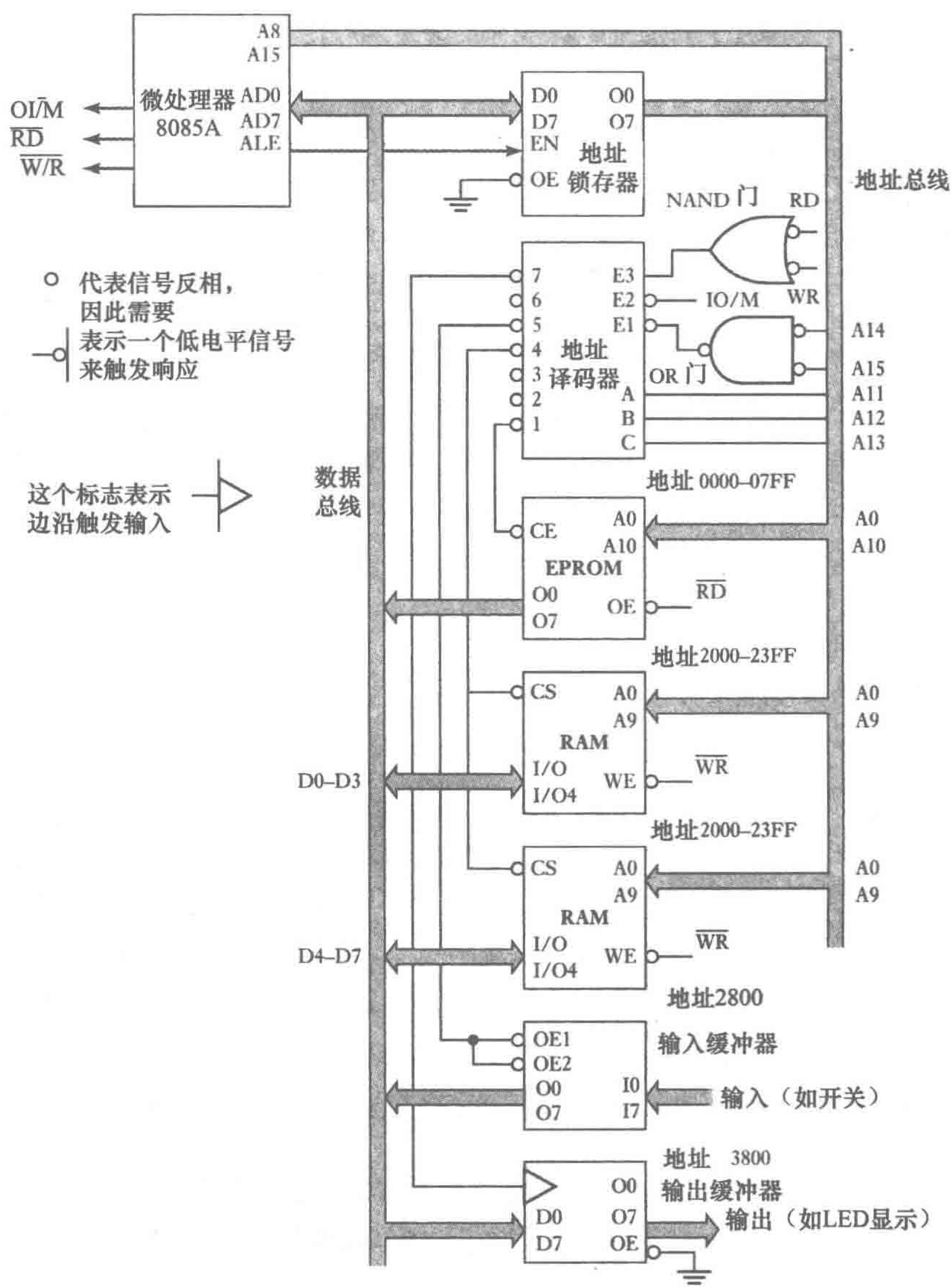


图10-7　Intel 8085A系统

1. 地址锁存器

地址锁存允许输出（ALE）向外部硬件提供输出信号，表示 AD0 ~ AD7 何时包含地址，何时包含数据。当 ALE 为高电平时，它使能锁存器锁存由 A0 ~ A7 传送过来的低位地址。因此当 ALE 变为低电平时，数据可以从微处理器输出，而低位地址仍然锁存在 74LS373 中。高位地址通过始终有效的 A8 ~ A15 来传送。这样，完整的地址由锁存器提供的低位部分和微处理器地址总线的高位部分共同提供。

2. 地址译码器

74LS138 是 3 - 8 译码器，它在 8 个输出口中的一个口提供一个低电平有效的输出。选择哪个输出口取决于 A、B 和 C 三个输入口的信号。在该译码器能实现这样的选择之前，E1 和 E2 必须接低电平，E3 接高电平来使能。

3. EPROM

地址位 A11、A12、A13 和 A14 是用于选择对哪个设备进行编址。剩下的地址位 A0 ~ A10 作为地址线使用。因此 EPROM 可提供 $2^{11}=2048$ 个地址。这就是 Intel 2716 EPROM 的存储容量。当微处理器读 0000 ~ 07FF 的地址或通过 O0 ~ O7 输出 8 位数据时，该 EPROM 都会被选择。输出选通信号 OE 和微处理器的读选通信号连接，从而保证只能从该 EPROM 读数据。

4. RAM

图 10-7 中可以看出系统使用了两个 RAM 芯片，每个为 1K × 4，它们一起提供 8 位宽的存储器。两个芯片共用 A0 ~ A9 的地址位，用于存储器选通。一个芯片提供数据 D0 ~ D3，另一个提供数据 D4 ~ D7。通过 10 个地址位，我们共有 2000 ~ 23FF 的 $2^{10}=1024$ 个地址。其写选通信号 $\overline{WR}$ 决定了 RAM 是读取状态还是写入状态。若 $\overline{WR}$ 为低电平，则所选择的 RAM 地址正在被写入；如果为高电平，则正在被读取。

250 ~ 251

5. 输入缓冲器

当 OE1 和 OE2 为低电平时，输入缓冲器 74LS244 被设定为可以通过数据总线的二进制输入值。它可以通过 2800 ~ 2FFF 之间的任意地址访问，因此我们可以用地址 2800。该缓冲器是用来保证输入只给微处理器带来非常小的负载。

6. 输出锁存器

74LS374 是一个输出锁存器。它用于锁存微处理器的输出信号，使输出设备有足够的时间去读取，而同时微处理器可以执行程序中的其他指令。该输出锁存器分配了从 3800 ~ 3FFF 之间的地址范围，因此可以用 3800 对它编址。

图 10-8 是使用摩托罗拉 6800 微处理器的系统范例。它只有一个 RAM 芯片和一个可编程 I/O 端口。因为涉及的设备数量较少，所以该系统不需要地址译码器。对于并行输入/输出，可使用外设接口适配器（PIA）（见 13.4 节），而对于串行输入/输出，可使用异步接口适配器（ACIA）（见 13.5 节）。这些适配器可编程来处理输入/输出信号，并给出所需的数据缓冲。

1. RAM

地址线 A14 和 A15 与 RAM 芯片的使能输入相连。当这两根线同时为低电平时，微处理器和 RAM 之间进行通信。

2. ROM

地址线 A14 和 A15 与 ROM 芯片的使能输入相连。当这两根线同时为高电平时，ROM 被寻址。

3. I/O

地址线 A14 和 A15 与 PIA 和 ACIA 的使能输入相连。当 A15 为低电平，同时 A14 为高电平时，I/O 端口被选中。为了表示是哪个设备被使能，地址线 A2 为高电平时表示 PIA 使能，地址线 A3 为高电平表示 ACIA 使能。

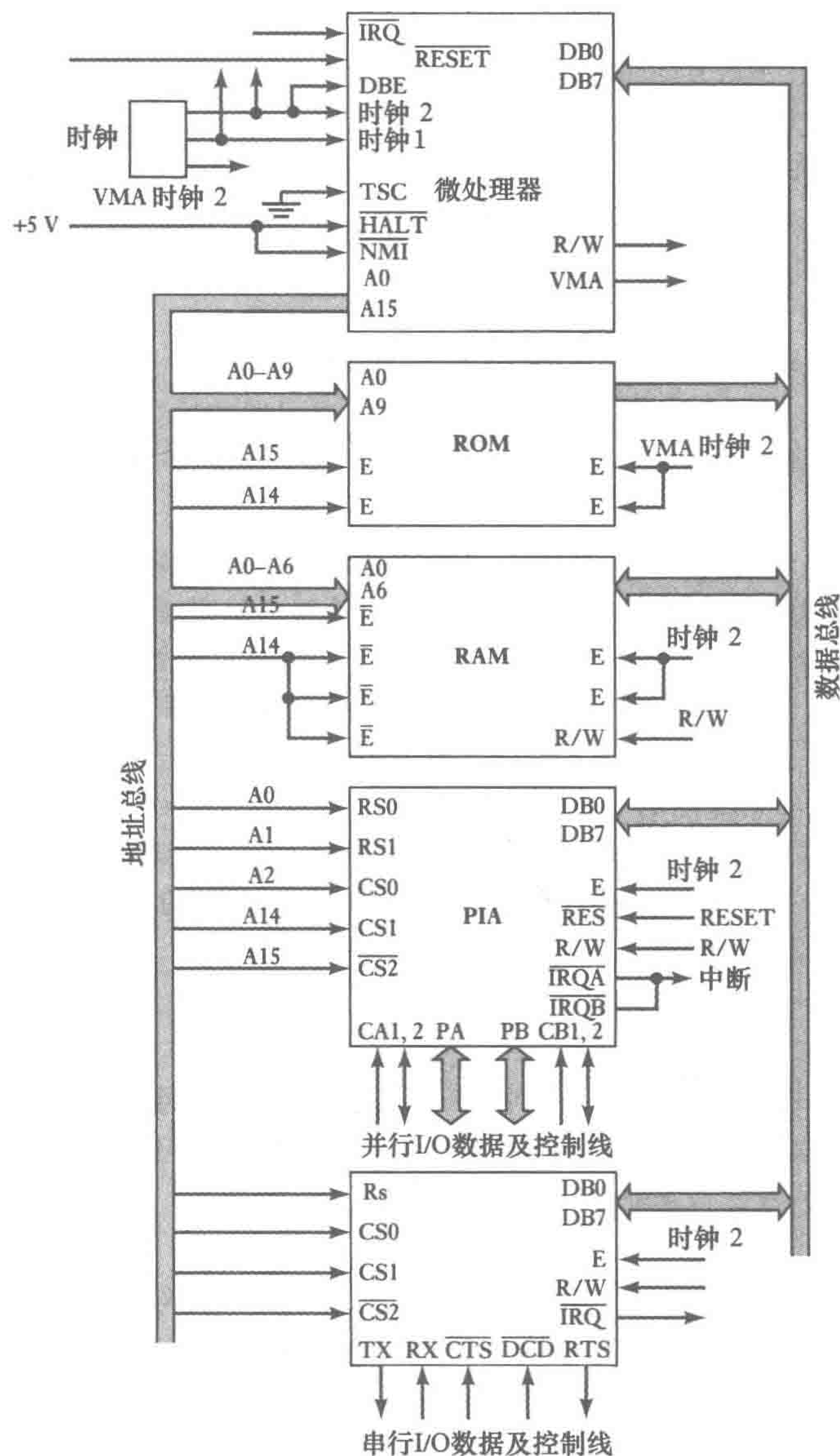


图 10-8 M6800 系统

10.3 微控制器

为了使微处理器能使系统用于控制，额外的芯片是必要的，比如存储程序和数据的存储设备和允许存储器与外部世界通信并接收信号的 I/O 端口。微控制器把微处理器、存储器、I/O 端口和其他外部设备（比如时钟）集成到一个芯片中。图 10-9 是微控制器的通用框图。

一般的微控制器有用于与输入/输出、电源、时钟和控制信号外部连接的引脚。用于输入/输出的引脚组成一个单元，称为 I/O 端口。通常这些端口有 8 根线，可以用来传输 8 位的数据字。两个 I/O 端口一起可用于传送 16 位的字，其中一个传送低 8 位，另一个传送高 8 位。该 I/O 端口可用于只输入、只输出或者由编程控制输入或输出。

摩托罗拉 68HC11、Intel 8051 和 PIC16C6x/7x 都是 8 位控制器的例子，因为它们的数据路径都是 8 位宽的。摩托罗拉 68HC16 是 16 位微控制器，而摩托罗拉 68300 是 32 位微控制器。

微控制器只有有限的 ROM 和 RAM，且广泛用于嵌入式系统。一个有单独存储器和 I/O 芯片的微处理器更适用于处理计算机系统中的信息。

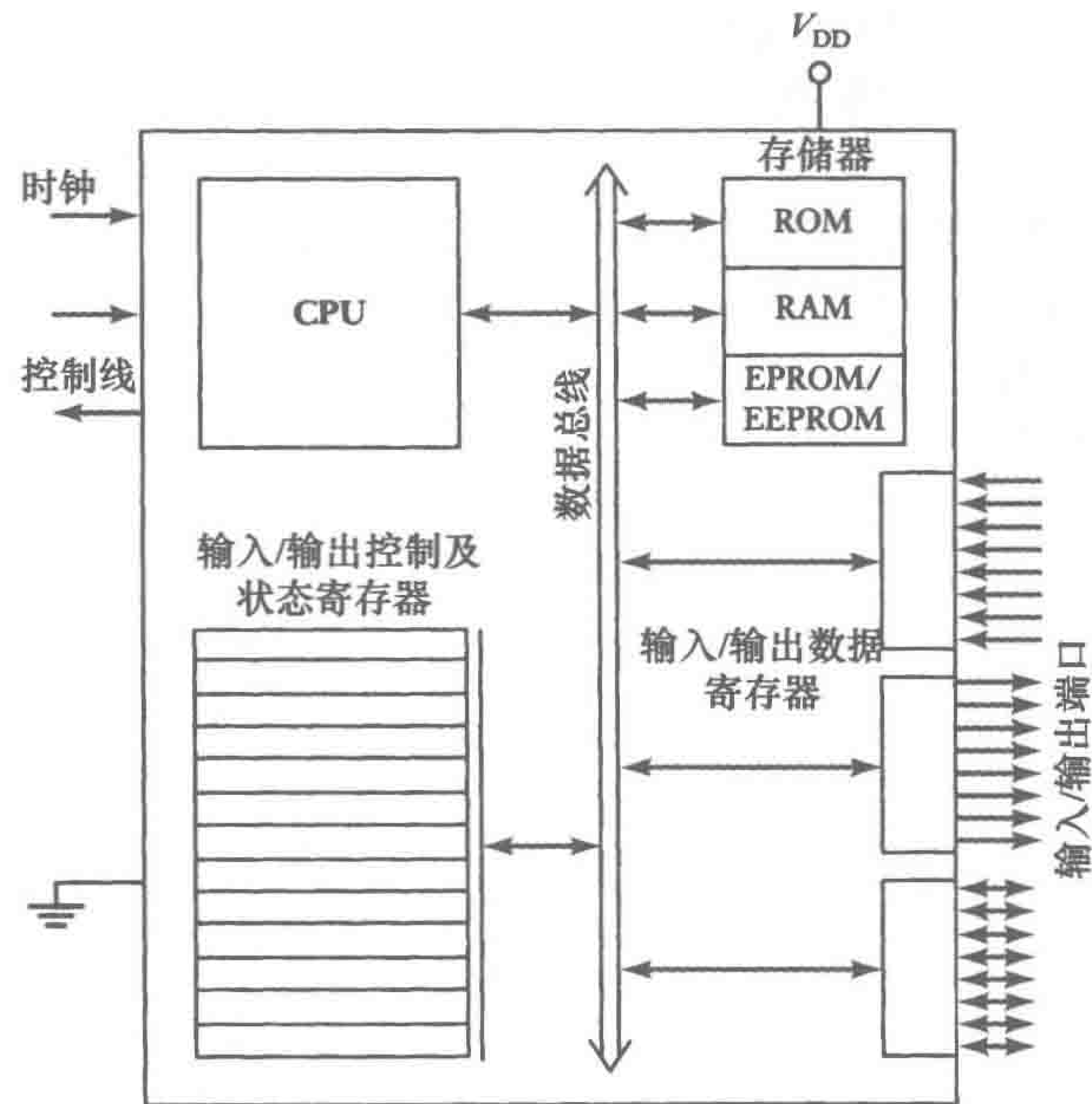


图 10-9　微控制器的框图

10.3.1　摩托罗拉 M68HC11

摩托罗拉提供两款基本的 8 位微控制器系列，68HC05 配备便宜的内核，而 68HC11 配备更高性能的内核。摩托罗拉 M68HC11 系列是基于摩托罗拉 6800 微处理器的（见图 10-10），广泛应用于控制系统。

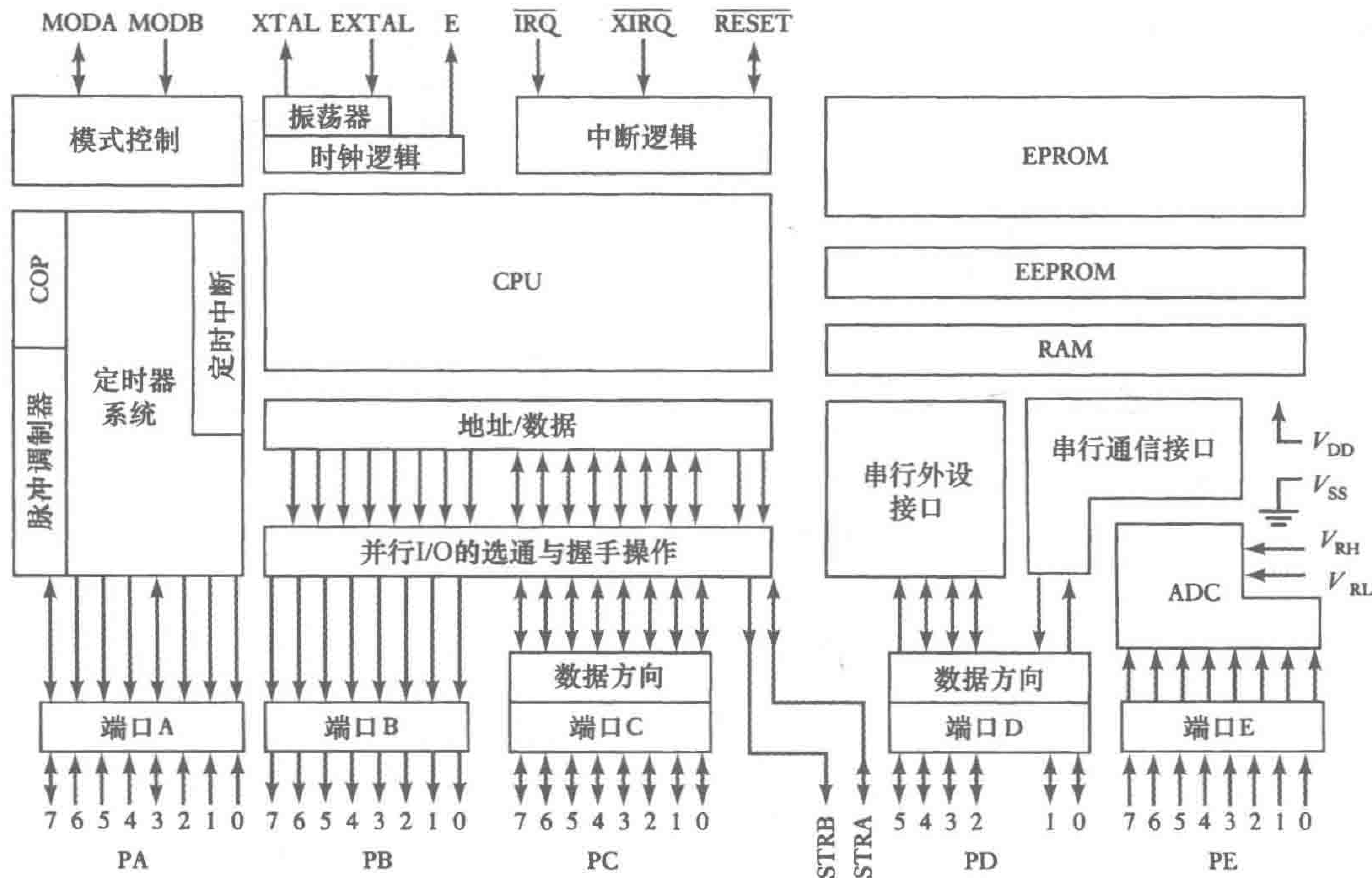


图 10-10　M68HC11 的框图

252～254

该系列中有许多不同的版本，它们在 RAM、ROM、EPROM、EEPROM 和配置寄存器的特征上有所不同。例如，68HC11A8 版本有 8K ROM，512 字节的 EEPROM，256 字节的 RAM，一个 16 位的定时器系统，一个同步串行外设接口，一个异步非返回零（non-return-to-zero）串行通信接口，一个 8 通道 8 位 A/D 转换器用于模拟输入，以及 A、B、C、D、E 五个端口。

1. 端口 A

端口 A 有 3 根输入线、4 根输出线和 1 根输入/输出线。端口 A 的数据寄存器的地址为 \$1000（见图 10-11），由一个地址为 \$1026 的脉冲累加器控制寄存器（见图 10-12）控制端口 A 中每一位的功能。该端口同样提供对微控制器内部定时器的访问，其中 PAMOD、PEDGE、RTR1 和 RTRO 位可控制脉冲累加器和时钟。

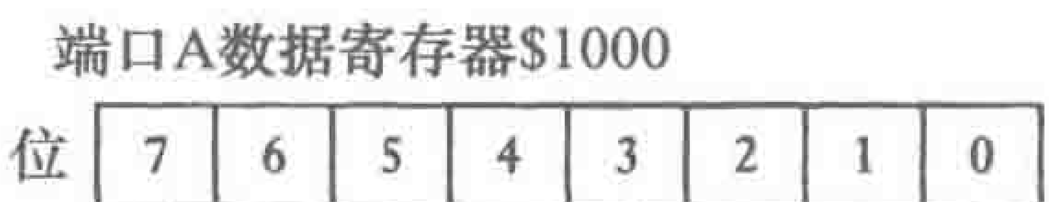


图 10-11　端口 A 寄存器

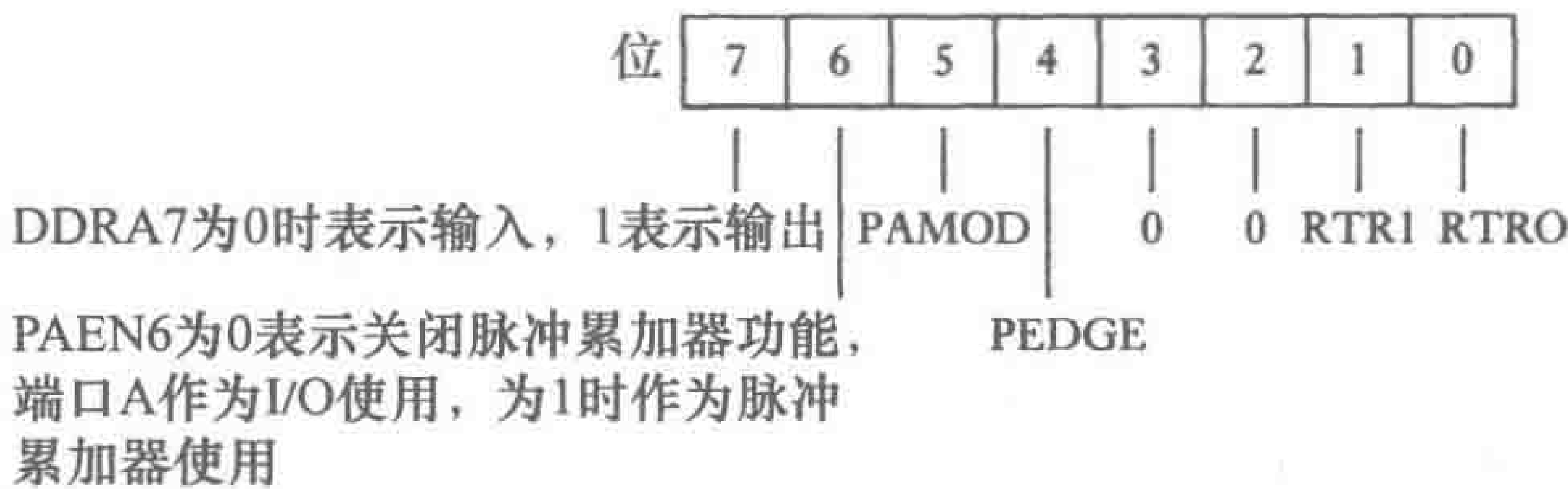


图 10-12　脉冲累加器控制寄存器

2. 端口 B

端口 B 只用于输出，有 8 根输出线（见图 10-13）。输入数据不能放入端口 B 的引脚。它的数据寄存器的地址为 \$1004。为了输出数据，数据必须要写入该存储器的地址中。

3. 端口 C

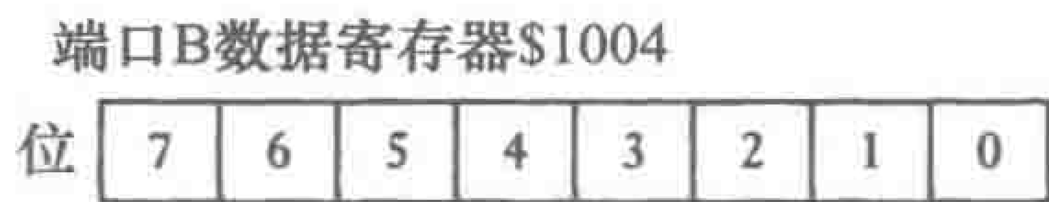


图 10-13　端口 B 寄存器

端口 C 可用于输出或输入，通过地址为 \$1003 的数据寄存器来读取和写入数据（见图 10-14）。它的方向受地址为 \$1007的端口数据方向寄存器的控制。该寄存器的 8 位和端口 C 的 8 位一一对应，决定这些数据线是输入还是输出。当时数据方向寄存器被置 0 时，为输入；当被置 1 时，为输出。线 STRA 和 STRB（当它们以单芯片模式操作时）与端口 B 和端口 C 有关，用于与这些端口的握手信号。这些线控制数据传输的时间。在地址为 \$1002 的并行 I/O控制寄存器 PIOC 包含了控制握手模式、极性和握手信号触发边沿的控制位。

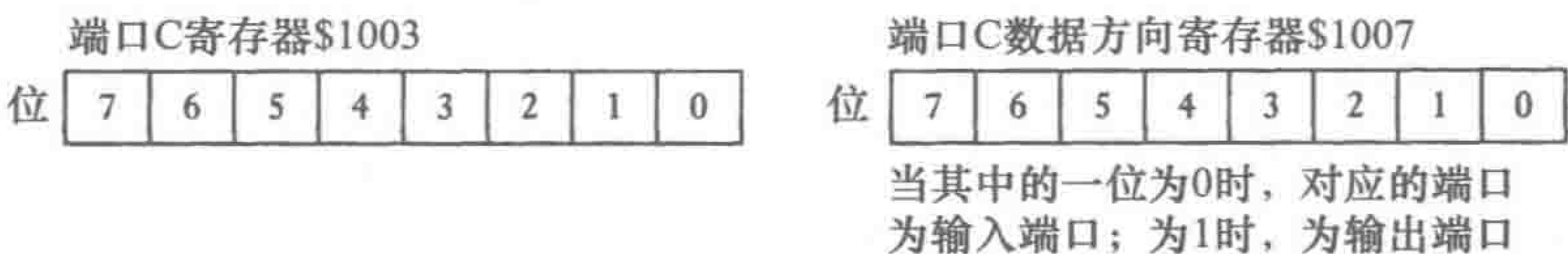


图 10-14　端口 C 寄存器

4. 端口 D

端口 D 有 6 根可输入可输出的数据线，它有一个地址为 \$1008 的数据寄存器（见图 10-15）。数据传输方向受地址为 \$1009 的端口数据方向寄存器的控制。当该寄存器的对应位被置 0 时，为输入；当被置 1 时，则为输出。其串行通信接口是一个能提供调制解调器和终端兼容的串行通信异步系统。串行外设接口是一个专门用于微控制器和外设部件之间通信的高速同步系统。

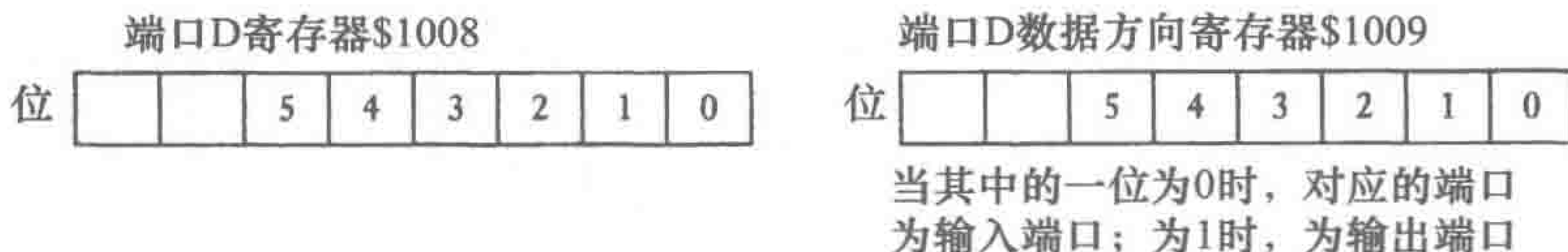


图 10-15　端口 D 寄存器

5. 端口 E

端口 E 是一个 8 位输入端口（见图 10-16）。它可用做通用输入端口，也可用于内部 A/D 转换器的模拟输入。两个输入 V_{RH}和 V_{RL}提供 A/D 转换器的参考电压。端口 E 的数据寄存器的地址是 $1002。



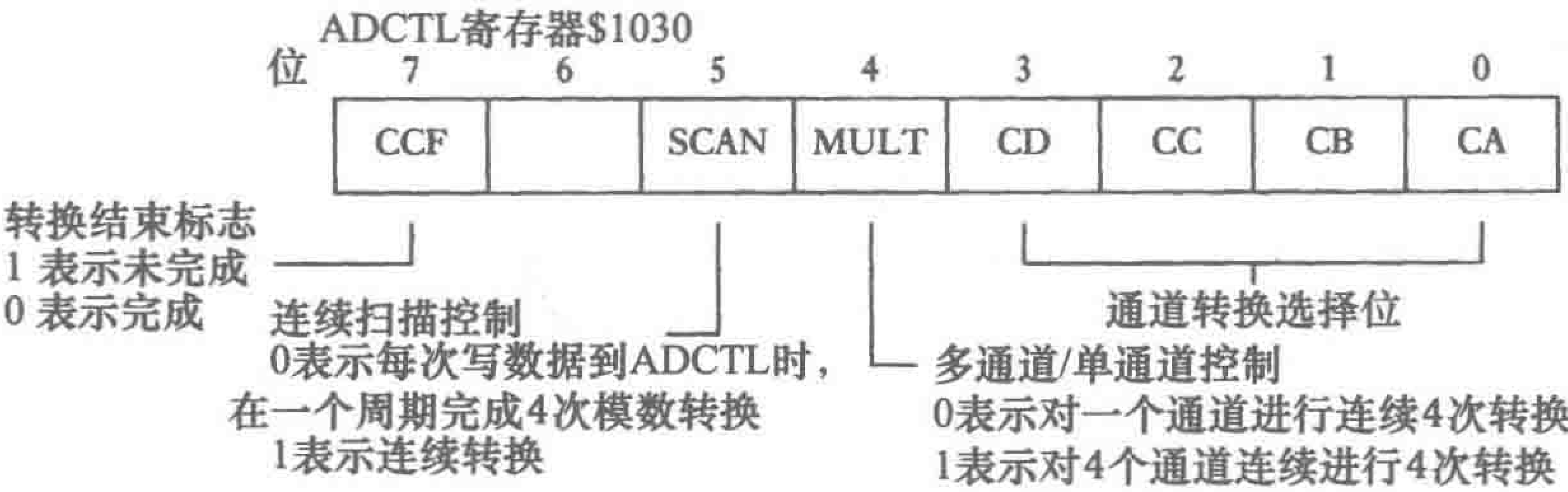
图 10-16　端口 E 寄存器

68HC11 有一个内部 A/D 转换器，端口 E 的 0、1、2、3、4、5、6 和 7 位是模拟输入引脚。V_{RH}和 V_{RL}提供 A/D 转换器所需要的参考电压，其中参考高电压 V_{RH}不能低于 V_{DD}，即 5V；而参考低电压 V_{RL}不能低于 V_{SS}，即 0V。A/D 转换器使用前必须先被使能，这是通过对 OPTION 寄存器中的 AD 启动（ADPU）控制位（见图 10-17）置 1 来实现的，即第 7 位置 1。第 6 位用于选择 A/D 转换器的时钟源。A/D 转换器启动后至少需要 100μs 才能稳定。



图 10-17　OPION 寄存器

在启动并稳定后，通过对 A/D 控制/状态寄存器（ADCTL）进行写操作可以初始化 A/D 转换器（见图 10-18）。这就涉及选择通道和操作模式。然后转换会在一个时钟周期后启动。例如，若通过设定 MULT = 0 来选择单通道模式，则在由 CD ~ CA 位选择的通道上实现 4 次连续的 A/D 转换，其转换结果放入 A/D 结果寄存器 ADR1 ~ ADR4。



MULT = 0

CD	CC	CB	CA	通道转换
0	0	0	0	PE0
0	0	0	1	PE1
0	0	1	0	PE2
0	0	1	1	PE3
0	1	0	0	PE4
0	1	0	1	PE5
0	1	1	0	PE6
0	1	1	1	PE7

MULT = 1

				结果寄存器			
CD	CC	CB	CA	ADR1	ADR2	ADR3	ADR4
0	0	×	×	PE0	PE1	PE2	PE3
0	1	×	×	PE4	PE5	PE6	PE7

图 10-18　ADCTL 寄存器

6. 模式

MODA 和 MODB 是 2 个可用于强制微控制器在启动后进入 4 种模式之一的引脚，这 4 种模式为特殊引导（special bootstrap）、特殊测试（special test）、单芯片（single chip）和扩展（expanded）：

MODB	MODA	模式
0	1	特殊引导
0	1	特殊测试
1	0	单芯片
1	1	扩展

255 ~ 257

在单芯片模式下，除了外部时钟源和复位电路外，微控制器是完全独立的。对于这种模式，微控制器对某些应用可能没有足够的资源，比如存储空间，因此可以使用扩展模式，这样地址的数量可增加。端口 B 和 C 可以提供地址、数据和控制总线。端口 B 作为高 8 位地址线，而端口 C 作为低 8 位地址线和数据线复用的引脚。引导模式允许生产厂家向特殊的 ROM 加载特殊的程序供 M68HC11 用户使用。当微控制器设定为该模式时，将加载特殊的程序。而特殊测试模式主要是在摩托罗拉内部生产测试时使用。

在选择模式后，MODA 引脚变为决定一个指令是否开始执行的引脚。MODB 引脚可在常规电源断开时，有给芯片内部 RAM 提供电源的功能。

7. 振荡器引脚

振荡器系统引脚 XTAL 和 EXTAL 与内部振荡器连接。图 10-19 是可能用到的外部电路图。E 是总线时钟，它的振荡频率是振荡器频率的 1/4，用于同步外部事件。

8. 中断控制器

中断控制器使微控制器中断一个程序（见 13.3.3 节）。中断是指一个事件要求 CPU 停止正常的程序执行，转而去执行与该事件相关的服务。IRQ 和 XIRQ 两根线用于中断信号的外部输入。RESET 用于复位微控制器，并使系统有序地启动。该引脚的状态可通过内部或外部进行设置。当检测到一个复位状态时，将引脚信号设为 4 个时钟周期的低电平。如果在接下来的两个周期内仍然为低电平，则认为已产生了一个外部复位信号。如果检测到电源输入 V_{DD}上有正跳转，则发生上电复位。这提供了 4064 个周期的延迟。如果在上电延迟时间的最后，复位引脚仍然为低电平，则微控制器保持休眠状态直至该引脚变为高电平。

9. 定时器

M68HC11 配备了定时器系统。它具有一个自由运行的计数器，5 输出比较功能，捕捉外部事件发生时间的能力，一个实时的周期性中断和一个称为脉冲累加器的外部事件计数器。自由运行计数器，又称为 TCNT，是一个 16 位计数器。它在 CPU 复位后从 0000 开始连续地计数，并且不可被程序复位。任意时刻都可以读取它的值。计数器的来源是系统总线时钟，它的输出通过设置地址为 $1024 的 TMSK2 寄存器的 PR0 和 PR1 位为 0 或 1 来进行预分频（见图 10-20）。

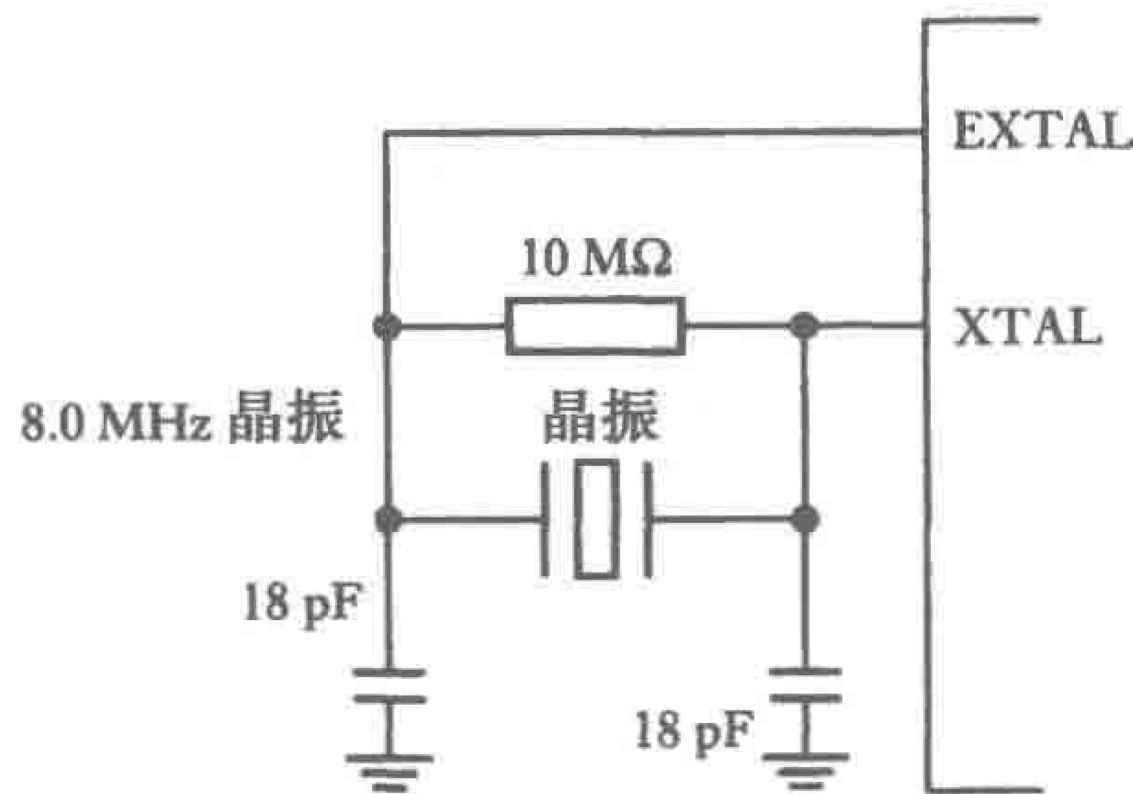


图 10-19 振荡器输出

地址为1024的定时器中断寄存器2 $1024

位	7	6	5	4	3	2	1	0
							PR1	PR0

分频因子

			每计一个数	
			总线频率	
PR1	PR0	分频因子	2MHz	1MHz
0	0	1	0.5ms	1ms
0	1	4	2ms	4ms
0	0	8	4ms	8ms
1	1	14	8ms	16ms

图 10-20 TMSK2 寄存器

当达到预设的计数值时，输出比较功能允许指定输出发生的时间，即定时器计数。输入捕捉系统能捕捉输入发生时计数器的值，因此可以捕捉输入发生的确切时刻。脉冲累加器可以设置成为一个事件计数器，对外部时钟脉冲计数，或者设置成一个门控时间累加器，存储由于计数器被使能和一段时间后被禁用所产生的特定时间间隔内的脉冲数。地址为 $1026 的脉冲累加器控制寄存器 PACTL（见图 10-12）用于选择操作的模式。其中 PAEN 置 0 可以禁用脉冲累加器，置 1 可使能累加器。PAMOD 置 0 为事件计数器模式，置 1 为门控计时模式。PEDGE 置 0 可使脉冲累加器在事件计数器模式下响应信号的下降沿，置 1 响应上升沿。在门控模式下，PEDGE 置 0 会导致计数器在端口 A 第 7 位为 0 时被禁用，而在端口 A 第 7 位为 1 时进行累加；PEDGE 置 1 使得计数器在端口 A 第 7 位为 1 时被禁用，而在端口 A 第 7 位为 0 时进行累加。

10. COP

另一个定时器功能为 COP，即计算机操作正常功能。如果某个操作没有在其合理的时间内结束，则该定时器会超时并复位系统（见 16.2 节）。通常它称为**看门狗定时器**。

11. PWM

脉宽调制（PWM）用于控制直流电动机的转速（见 3.6 节和 9.5.3 节），它是通过改变一个方波信号高电平的时间量，从而改变信号的平均值来实现的。方波信号可由微控制器每半个周期输出一次来产生。但是，M68HC11 的某些版本具有 PWM 模块，因此 PWM 模块被初始化和使能后，就可以自动地输出 PWM 波形。

258 ~ 259

从上面的论述中可以明显看出，在微控制器使用前必须对其进行初始化，即对寄存器的某些位进行设置，从而实现所需要的功能。

10.3.2　Intel 8051

另外一个常见的微控制器系列为 Intel 8051，图 10-21 为它的引脚连接和结构。8051 有 4 个并行 I/O 端口——P0、P1、P2 和 P3。P0、P2 和 P3 还有其他功能。8051AH 版本配备 4K ROM，128 字节 RAM，两个定时器和 5 个中断源的中断控制。

1. I/O 端口

P0、P1、P2 和 P3 的地址分别为 80H、90H、A0H 和 B0H（注意 Intel 的地址后面的 H 或 h 是用于表示地址是十六进制的）。当某个端口用做输出端口时，数据被放入对应的特殊功能寄存器。当某个端口用做输入端口时，必须首先对其写入 FFH。所有端口是位寻址的。因此，举例来说，我们可以用端口 P0 的第 6 位来控制一个电动机的启动或停止，用第 7 位来控制泵的打开或关闭。

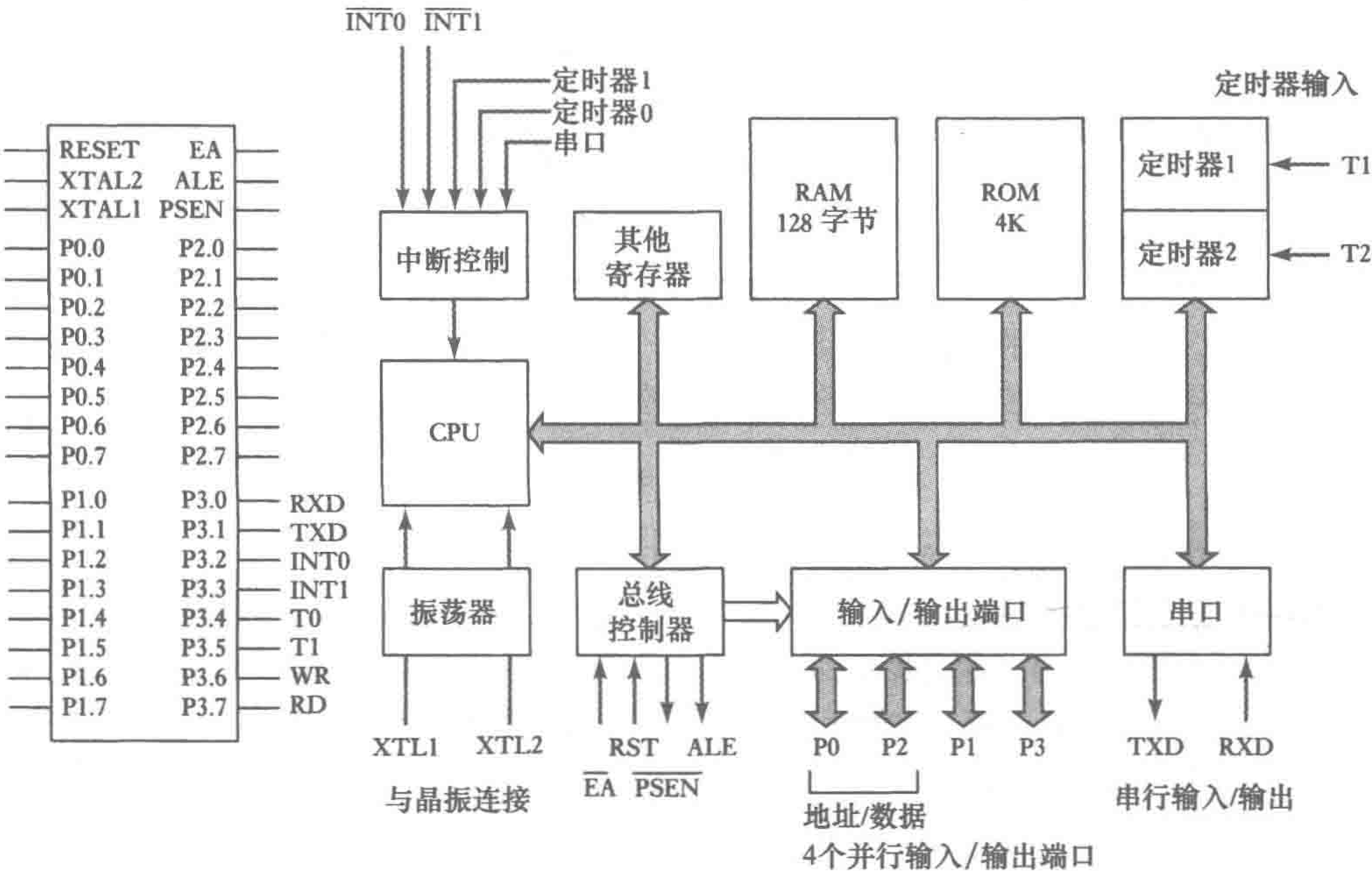


图 10-21　Intel 8051 引脚连接和结构

260

端口 P0 可用做输出端口或者输入端口。另外，它可用做多路地址和数据总线去访问外部存储器。端口 P1 可以用做输出端口或输入端口。端口 P2 可以用做输出端口或输入端口。另外，它还可以用做地址总线的高位去访问外部存储器。端口 P3 可以用做输出端口或输入端口。另外它可用做特殊用途的 I/O 端口。端口 P3 的其他功能包括中断和定时器的输出、串口的输入/输出和与外部存储器接口的控制信号。RXD 是串行输入端口，TXD 是串行输出端口，INT0 是外部中断

0，INT1 是外部中断 1，T0 是定时器/计数器 0 的外部输入，T1 是定时器/计数器 1 的外部输入，WR 是外部存储器写选通，RD 是外部存储器读选通。**选通**描述的是用于使能或禁止某种特殊功能的连接。端口 P0 可用做输出端口或输入端口。另外它可用于访问外部存储器。

2. ALE

在访问外部存储器时，地址锁存使能（ALE）提供输出脉冲来锁存地址信号的低 8 位字节。这样就可以使用 16 位的地址。图 10-22 描述了该过程。

3. PSEN

片外程序存储器选通控制信号（PSEN）引脚是外部程序存储器的读信号引脚，低电平有效，它与外部 ROM 或 EPROM 的输出使能引脚相连。

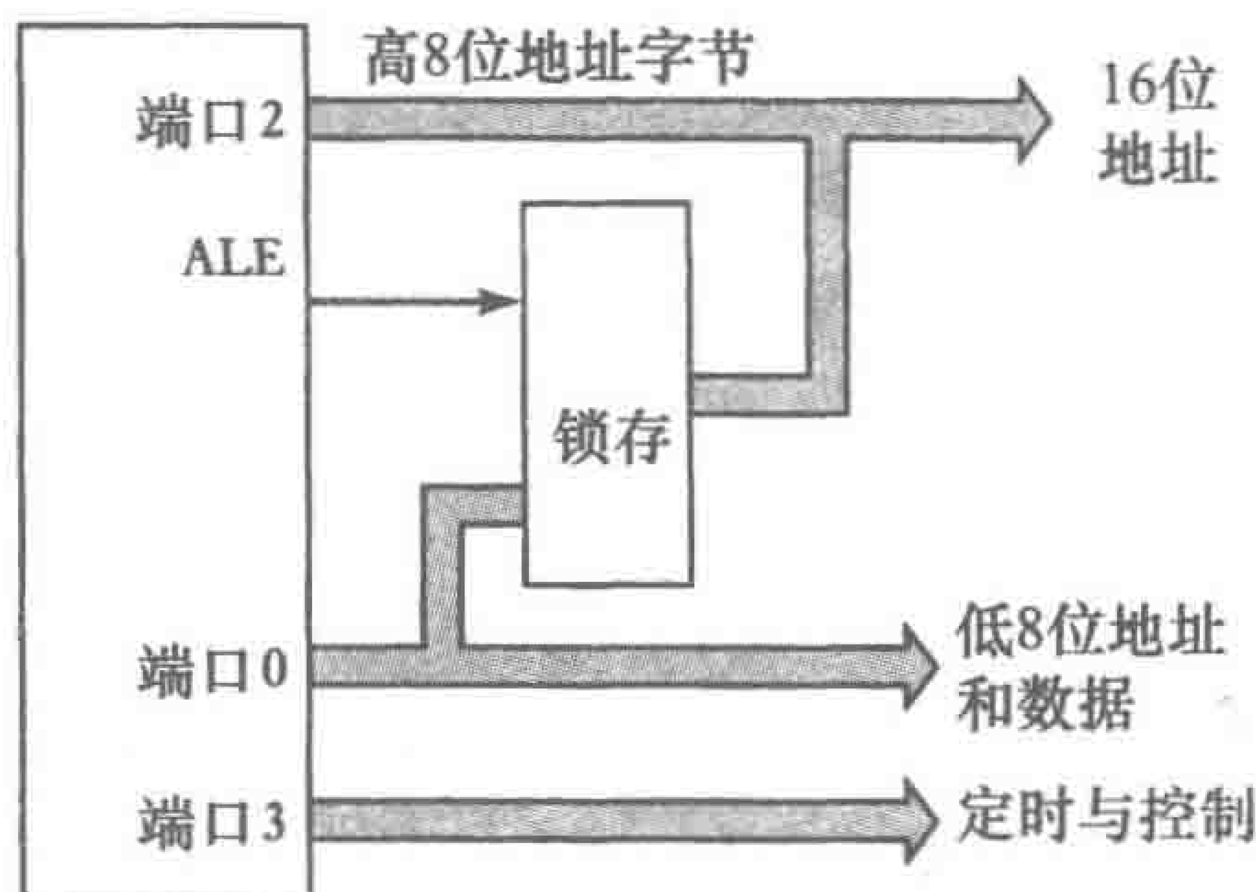


图 10-22　ALE 的使用方法

4. EA

外部访问（EA）引脚为低电平时，微处理器只能访问外部存储器；高电平时，微处理器根据地址自动访问内部或外部代码。因此，除非 EA 为低电平，否则 8051 被复位时，程序计数器 PC 将会从 $0000 开始指向内部程序存储器的第一条程序指令。然后 CPU 使 PSEN 处于低电平来使能外部程序存储器。该引脚也用于有 EPROM 的微控制器接收编程电压，这样就可以对 EPROM 进行编程。

261

5. XTAL1，XTAL2

这两个引脚用于连接晶振或外部振荡器。图 10-23 描述了怎样使用晶振。最常用的晶振频率为 12MHz。

6. RESET

对该引脚输入至少两个机器周期的高电平信号将会复位微控制器，即为有序的系统启动添加了一个条件。

7. 串行输入/输出

向地址为 99H 的串行数据缓冲器 SBUF 写入数据，用于数据发送；存取接收的数据时将读取 SBUF。地址为 98H 的可位寻址的串口控制寄存器 SCON 负责控制不同模式的操作。

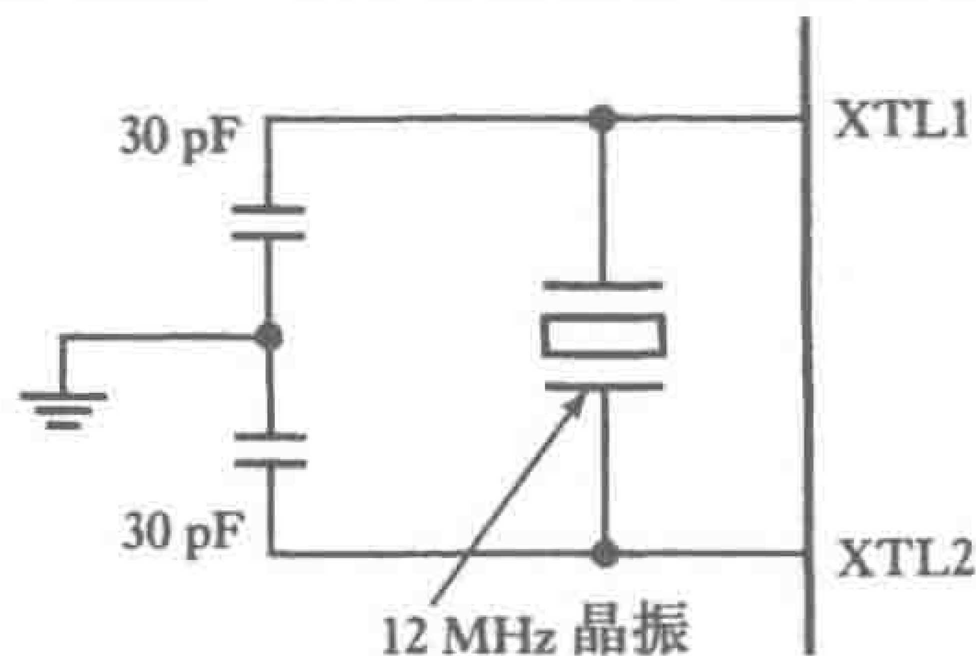


图 10-23　晶振

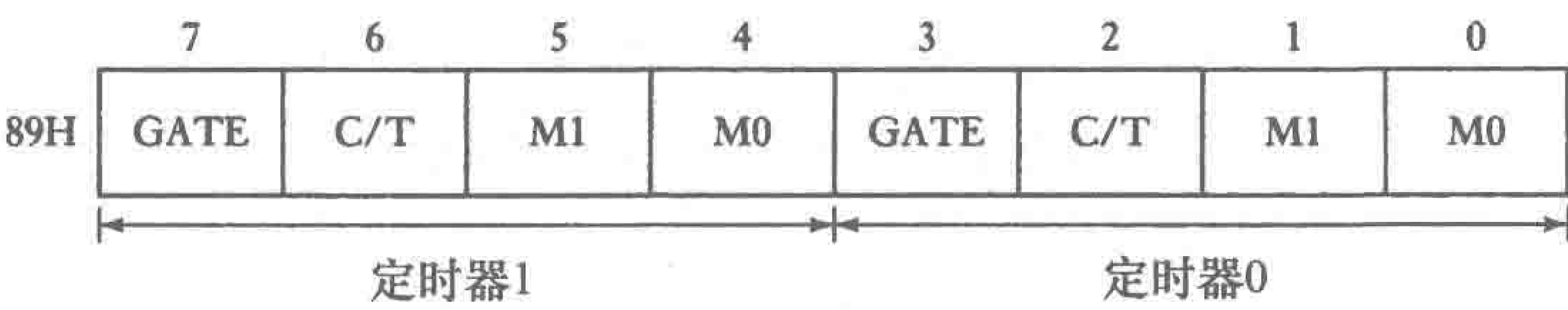
8. 定时器

地址为 89H 的定时器模式寄存器 TMOD 用于设定定时器 0 和 1 的工作模式（见图 10-24）。它是一个整体，不能对位寻址。定时器控制寄存器 TCON（见图 10-25）包含了定时器 0 和 1 的状态和控制位。它的高 4 位用于启动和关闭定时器并发出定时器溢出信号，而其 4 位和定时器无关，但用于检测和初始化外部中断。

每个定时器计数的信号源是通过 C/T 位设置的。如果该位为低电平，则定时器的信号源为系统时钟除以 12；如果该位为高电平，则对外部信号源的输入进行计数。可以通过设置 TR0 或 TR1 为 1 来启动定时器，或设置为 0 来停止定时器。另外一种控制定时器的方法是把 GATE 位设置为 1，这时定时器受微控制器的 INT0 或 INT1 引脚变为 1 来控制。这样外部设备可以通过连接其中某个引脚来控制计数器的打开/关闭。

9. 中断

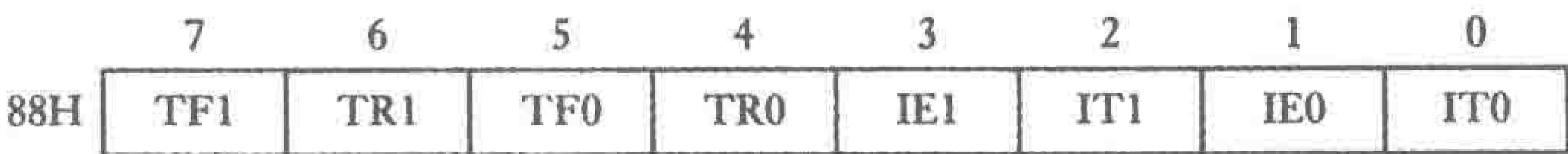
中断强制程序调用位于存储器中特定位置的子程序，通过写地址为 A8H 的中断允许寄存器 IE 可以允许中断（见图 10-26）。



Gate：0表示当TR0/TR1被置位时，定时器开始工作
　　1表示只有当INT0/INT1和TR0/TR1都为高时，定时器开始工作
C/T：计数器/定时器选择
　　0表示使用系统内部时钟，1表示使用TX0/TX1输入时钟
M0和M1设置模式

M1	M0	模式	
0	0	0	13位的计数器，TL0低5位，TH0的所有8位
0	1	1	16位计数器
1	0	2	8位自动重载定时器/计数器
1	1	3	TL0是由定时器0控制位控制的8位定时器/计数器 TH0是由定时器1控制位控制的8位定时器 定时器1关闭

图 10-24　TMOD 寄存器



TF0,TF1	定时器溢出标识。当定时溢出时由硬件置位，且当处理器调用中断处理程序时，由硬件清除该标志位
TR0,TR1	定时器运行控制位：1表示定时器打开，0表示定时器关闭
IE0,IE1	中断请求标志位：当有外部中断沿或检测到低电平信号时，由硬件将其置位。当中断被处理时，由硬件将其清除
IT0,IT1	软件设置中断类型：1表示下降沿触发，0表示低电平触发

图 10-25　TCON 寄存器

262

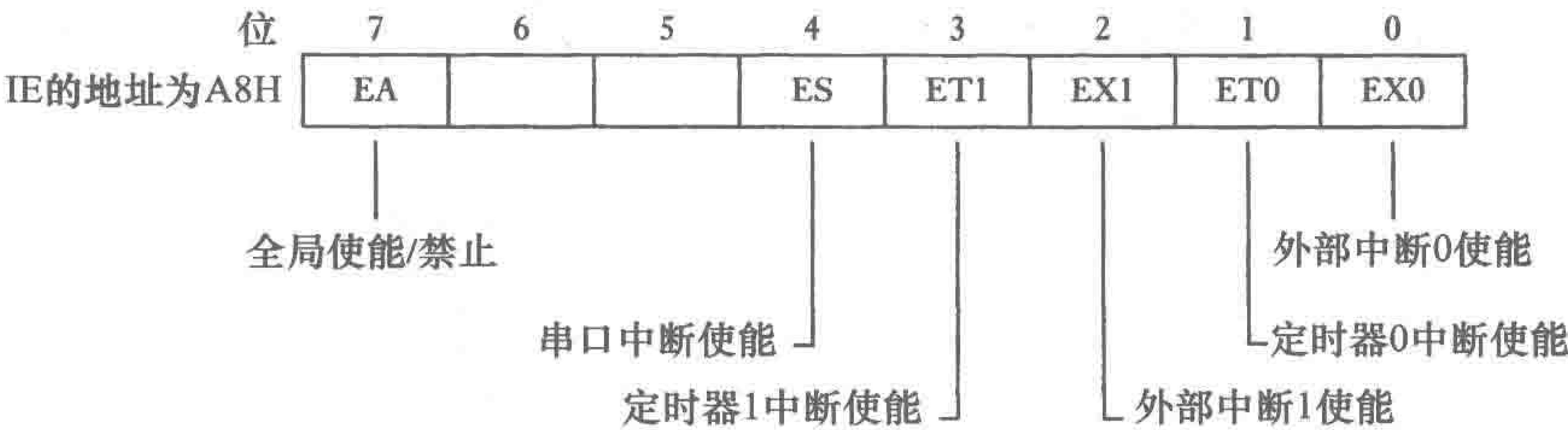


图 10-26　IE 寄存器

特殊功能寄存器是用于控制输入/输出的寄存器（见图 10-27），与上面的 IE 一样，它们位于 80～FF 的地址范围内。累加器 A（ACC）是数据操作的主寄存器，而 B 则是用于乘法和除法的寄存器。P0、P1、P2 和 P3 则是端口 0、1、2 和 3 的锁存寄存器。

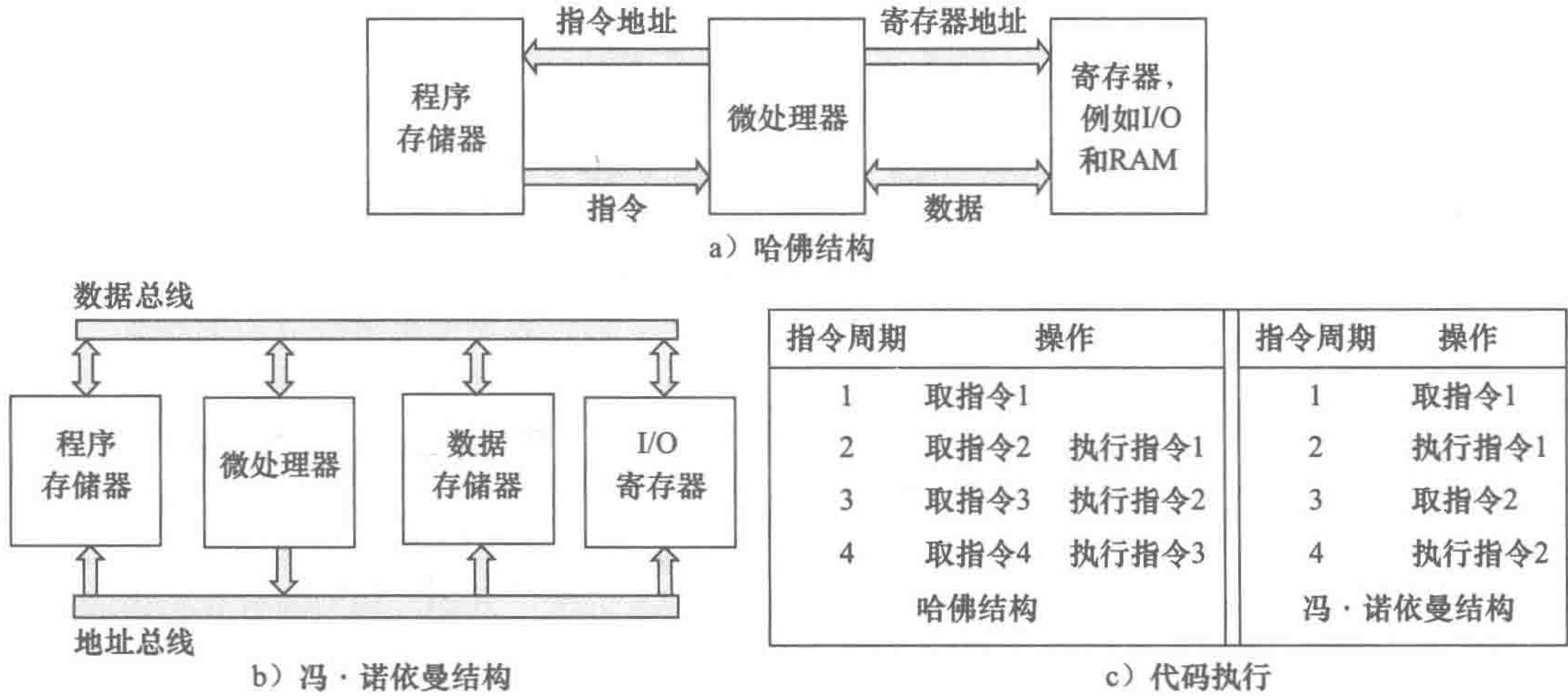
10.3.3　Microchip 微控制器

另外一个广泛应用的 8 位微控制器系列由 Microchip 提供。外设接口控制器（PIC）用于单芯片微控制器，它们使用的结构是**哈佛结构**。在这个结构下，获取程序指令的总线和访问数据的总线是分开的（图 10-28）。在本章中讨论的其他微处理器没有用分开的总线，因此，在下一条指令被接收到存储器中前，程序数据的获取必须等到读/写和输入/输出操作结束后才能进行。通过哈佛结构，程序指令在每个周期都可以获取而不需要等待，每条指令在取指周期后执行。哈佛结构使得在给定的时钟频率下能获得更快的执行速度。图

地址	寄存器	地址	寄存器
8D	TH1	F0	B
8C	TH0	E0	ACC
8B	TL1	D0	PSW
8A	TL0	B8	IP
89	TMOD	B0	P3
88	TCON	A8	IE
87	PCON	A0	P2
83	DPH	99	SBUF
82	DPL	98	SCON
81	SP	90	P1
80	P0		

图 10-27　寄存器

10-29 是 PIC16C74A 的某个版本和 16F84 微控制器的引脚图，而图 10-30 是该结构的基本形式。



a）哈佛结构

b）冯·诺依曼结构

c）代码执行

图 10-28

PIC16C74A

引脚	名称	引脚	名称
1	$\overline{MLCR}/V_{PP}$	40	RB7
2	RA0/AN0	39	RB6
3	RA1/AN1	38	RB5
4	RA2/AN2	37	RB4
5	RA3/AN3/V	36	RB3
6	RA4/T0CKI	35	RB2
7	RA5/$\overline{SS}$/AN4	34	RB1
8	RE0/$\overline{RD}$/AN5	33	RB0/INT
9	RE1/$\overline{WR}$/AN6	32	V_{DD}
10	RE2/$\overline{CS}$/AN7	31	V_{SS}
11	V_{DD}	30	RD7/PSP7
12	V_{SS}	29	RD6/PSP6
13	OSC1/CLKIN	28	RD5/PSP5
14	OSC2/CLKOUT	27	RD4/PSP4
15	RC0/T1OSO/T1CKI	26	RC7/RX/DT
16	RC1/T1OSI/CCP2	25	RC6//TX/CK
17	RC2/CCP1	24	RC5/SDO
18	RC3/SCK/SCL	23	RC4/SDI/SDA
19	RD0/PSPO	22	RD3/PSP3
20	RD1/PSP1	21	RD2/PSP2

PIC16F84A

引脚	名称	引脚	名称
1	RA2	18	RA1
2	RA3	17	RA0
3	RA4/T0CK1	16	OSC1/CLKIN
4	$\overline{MCLR}$	15	OSC2/CLKOUT
5	V_{SS}	14	V_{DD}
6	RB0/INT	13	RB7
7	RB1	12	RB6
8	RB2	11	RB5
9	RB3	10	RB4

PIC18F452

引脚	名称	引脚	名称
1	$\overline{MLCR}/V_{PP}$	40	RB7
2	RA0/AN0	39	RB6
3	RA1/AN1	38	RB5
4	RA2/AN2	37	RB4
5	RA3/AN3/V	36	RB3/CCP2
6	RA4/T0CKI	35	RB2/IN2
7	RA5/SS/AN4/LVDIN	34	RB1/INT1
8	RE0/$\overline{RD}$/AN5	33	RB0/INT0
9	–RE1/$\overline{WR}$/AN6	32	V_{DD}
10	RE2/$\overline{CS}$/AN7	31	V_{SS}
11	V_{DD}	30	RD7/PSP7
12	V_{SS}	29	RD6/PSP6
13	OSC1/CLKIN	28	RD5/PSP5
14	OSC2/CLKOUT/RA6	27	RD4/PSP4
15	RC0/T1OSO/T1CKI	26	RC7/RX/DT
16	RC1/T1OSI/CCP2	25	RC6//TX/CK
17	RC2/CCP1	24	RC5/SDO
18	RC3/SCK/SCL	23	RC4/SDI/SDA
19	RD0/PSPO	22	RD3/PSP3
20	RD1/PSP1	21	RD2/PSP2

图 10-29 PIC 引脚图

16C74 的基本特征如下所述（其他 PIC 微控制器有类似的功能）。

1. I/O 端口

引脚 2、3、4、5、6 和 7 是给双向 I/O 端口 A 使用的。和其他双向端口一样，信号可以通过端口寄存器读出或写入。信号的方向由 TRIS 方向寄存器控制，每个端口都有一个 TRIS 寄存器。TRIS 设置 1 时方向为读出，设置 0 时方向为写入（见图 10-31）。

图 10-30　PIC 16C74/74A

引脚 2、3、4 和 5 也可以用于模拟输入，引脚 6 可用于定时器 0 的时钟输入，而引脚 7 可作为同步串口的从动选择（见本节的后续部分）。

263~264

引脚 33、34、35、36、37、38、39 和 40 为双向 I/O 端口 B，其信号的方向由对应的 TRIS 方向寄存器控制。33 引脚也可以用做外部中断引脚。引脚 37、38、39 和 40 也可作为有电平改变中断的引脚。引脚 39 也可作为串行编程时钟，而引脚 40

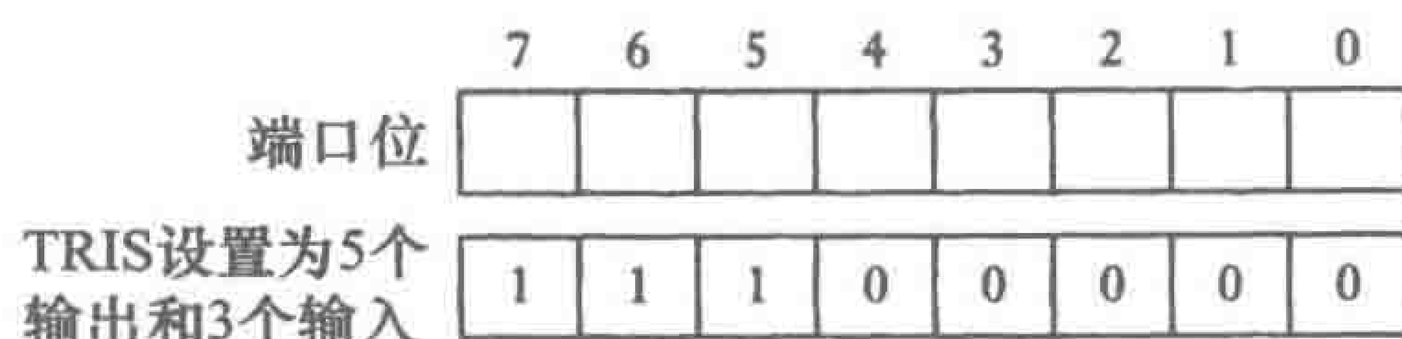


图 10-31　端口传输方向

可作为串行编程数据。

引脚15、16、17、18、23、24、25和26为双向I/O端口C，其信号的方向由对应的TRIS方向寄存器控制。引脚15可作为定时器1的输出或时钟输入。引脚16也可作为定时器1的晶振输入或者捕获2输入/比较2输出/脉宽调制2输出。

引脚19、20、21、22、27、28、29和30为双向I/O端口D，其信号的方向由对应的TRIS方向寄存器控制。

引脚8、9和10为双向I/O端口E，其信号的方向由对应的TRIS方向寄存器控制。引脚8也可用做并行从动（slave）端口或者模拟输入5的读控制。并行从动端口是一项有利于个人计算机接口电路设计的特征。在使用时，端口D和E专门用于此功能。

2. 模拟输入

端口A的2、3、4、5、7引脚和端口E的8、9、10引脚可用做模拟输入，是内部A/D转换器的输入。在进行转换前，必须对端口A的ADCON1和TRISA寄存器（端口E的TRISE寄存器）初始化来选择参考电压和输入通道。然后必须根据以下设定初始化ADCON0：

ADCON0位			模拟输入
5	4	3	
0	0	0	端口A，位0
0	0	1	端口A，位1
0	1	0	端口A，位2
0	1	1	端口A，位3
1	0	0	端口A，位5
1	0	1	端口E，位0
1	1	0	端口E，位1
1	1	1	端口E，位2

3. 定时器

微控制器有3个定时器：定时器0、定时器1和定时器2。定时器0是可以写入或者读取的8位计数器，它可用做对外部信号的变化计数，当要求的事件次数达到时产生中断。计数器的信号来源可以是内部的总线时钟或者外部的数字信号。计数器的来源选择由OPTION寄存器中的TOCS位决定（见图10-32）。

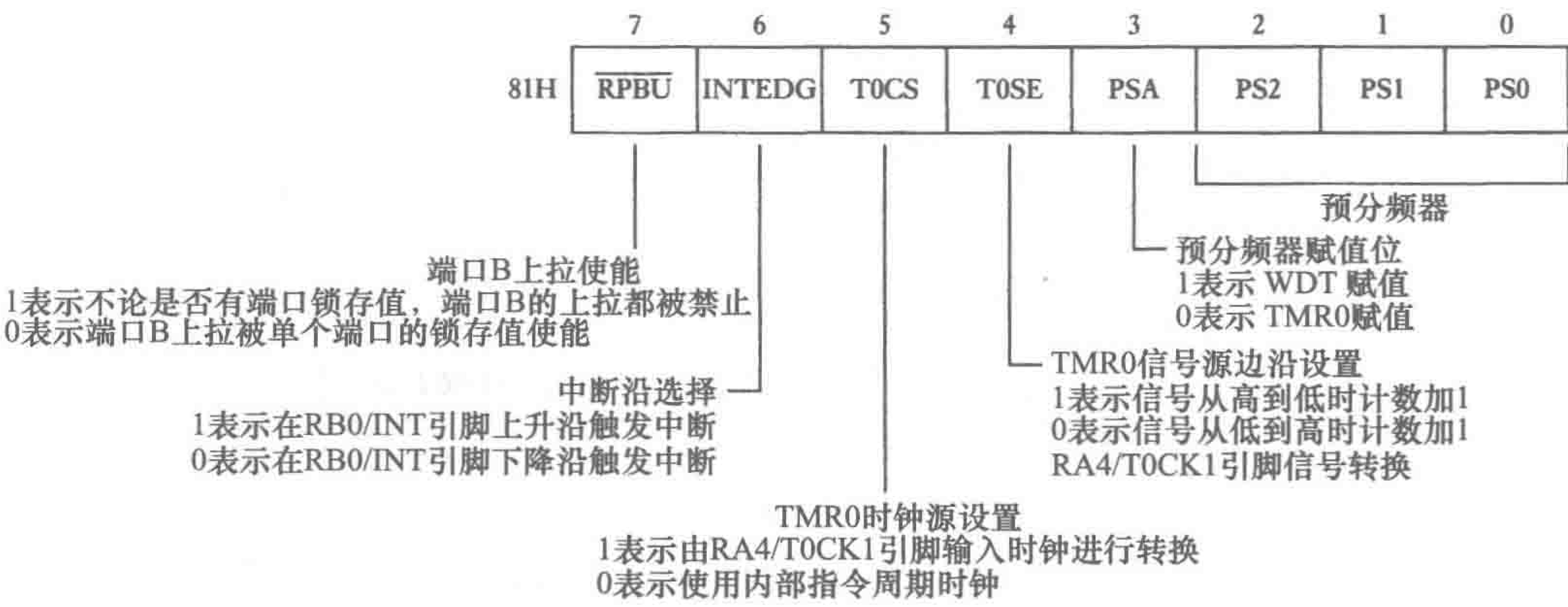


图10-32　OPTION寄存器

如果没有选择预分频器，则计数器每隔两个输入信号源周期递增一次。使用预分频器可以

使信号只在固定数目的时钟周期后才传送到计数器。以下是可能的分频率。当看门狗定时器被使能时，WDT 给出了被选择的分频系数。如果一个操作没在合理的时间内结束，它就会用于超时判断并复位系统，该时间默认值为 18ms。

265 ～ 267

预分频器位值			TMR0 分频比	WDT 分频比
PS2	PS1	PS0		
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

定时器 1 是具有最多功能的定时器，它可用于控制输入引脚上信号转换之间的时间，或者控制输出引脚上的精确转换时间。当定时器 1 被设定为捕获或比较模式时，它使能微控制器去控制引脚 17 上的输出定时。

定时器 2 可用于控制 PWM 输出的周期。PWM 输出由引脚 16 和 17 提供。

4. 串行输入/输出

PIC 微控制器包含一个同步串口（SSP）模块和一个串行通信接口模块（SCI）。引脚 18 的其他功能是 SPI 串行外设接口模式和 I^2C 模式下的同步串行时钟输入或输出。I^2C 总线提供两线制的双向接口，可供其他的芯片使用。它还可用于连接主从微控制器。UART，即通用异步收发器，可以用于给个人计算机建立一个串口。

5. 并行从动端口

并行从动端口使用端口 D 和 E，使微处理器提供一个与个人计算机的接口。

6. 晶振输入

引脚 13 用于晶振输入或外部时钟源输入；引脚 14 为晶振输出。图 10-33a 是可能用到的精确控制频率的电路图。图 10-33b 是可能用到的低成本频率控制电路，若需要 4MHz 的频率，则我们可以使用 $R=4.7\text{k}\Omega$ 和 $C=33\text{pF}$。内部时钟频率为振荡器频率的 1/4。

7. 主清除和复位

引脚 1 为主清零 MCLR，即复位输入，它处于低电平时复位控制器，并使系统有序地启动。当检测到 V_{DD} 增加时，则产生一个上电复位（POR）脉冲来提供一个固定的超时延迟，从而使处理器保持复位状态。若 V_{DD} 低于某一电压超过一定时间，则掉电复位将被激活。看门狗定时器是另一种复位方式，某个操作没在合理的时间内结束时，它超时并复位微控制器。

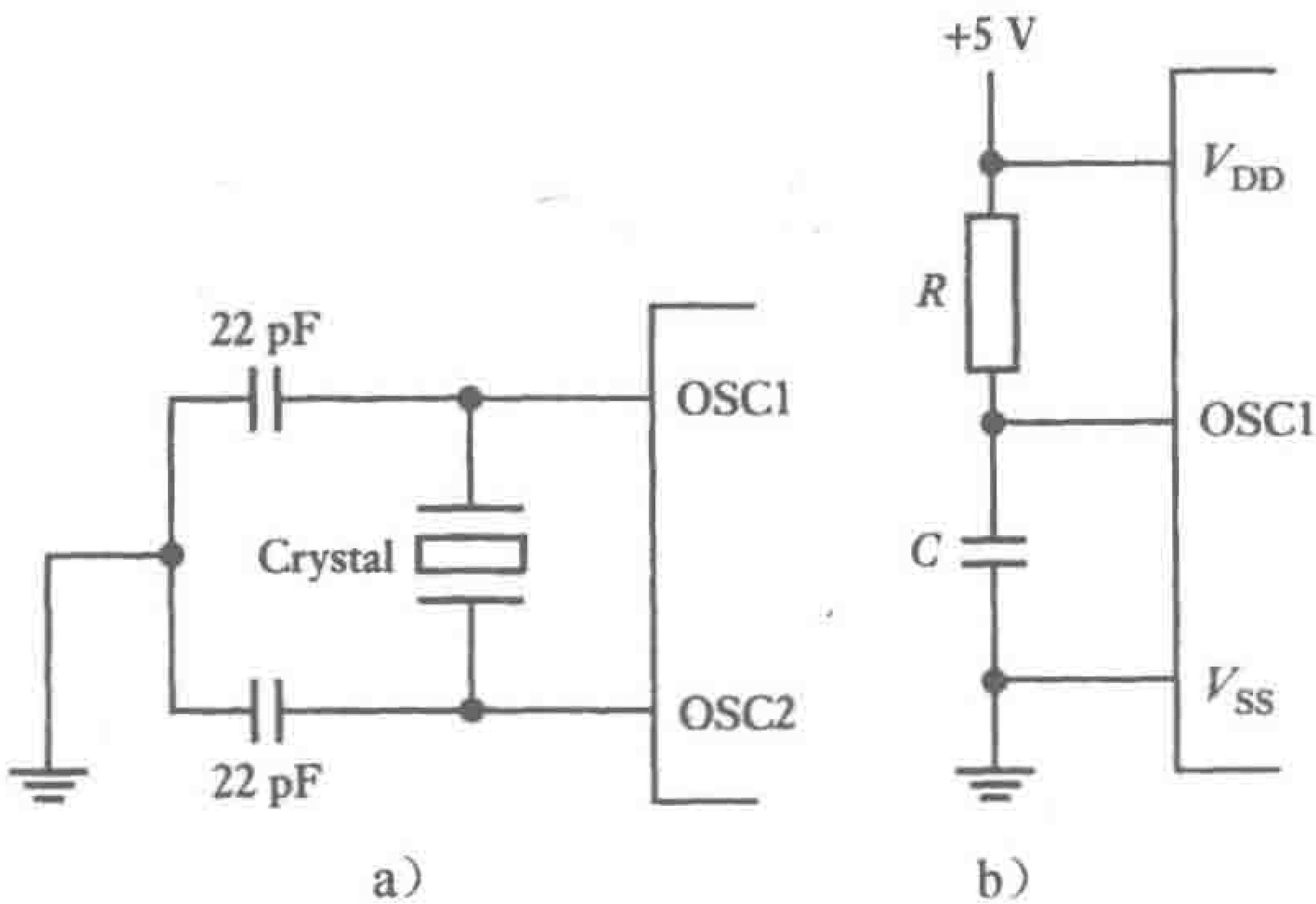


图 10-33　频率控制

和上述有关的这些寄存器一样，**专用寄存器**（见图 10-34）用于输入/输出控制。PIC16C73/74 的寄存器分为两组（bank），在选中某个寄存器前，一定要通过设置状态寄存器的一位（见图 10-35）来选择其中一组。

文件地址	组0	组1	文件地址
00h	INDF	INDF	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	PORTC	TRISC	87h
08h	PORTD	TRISD	88h
09h	PORTE	TRISE	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh	PIR2	PIE2	8Dh
0Eh	TMR1L	PCON	8Eh
0Fh	TMR1H		8Fh
10h	T1CON		90h
11h	TMR2		91h
12h	T2CON	PR2	92h
13h	SSPBUF	SSPADD	93h
14h	SSPCON	SSPSTAT	94h
15h	CCPR1L		95h
16h	CCPR1H		96h
17h	CCP1CON		97h
18h	RCSTA	TXSTA	98h
19h	TXREG	SPBRG	99h
1Ah	RCREG		9Ah
1Bh	CCPR2L		9Bh
1Ch	CCPR2H		9Ch
1Dh	CCPR2CON		9Dh
1Eh	ADRES		9Eh
1Fh	ADCON0	ADCON1	9Fh
20h 7Fh	通用寄存器	通用寄存器	A0h FFh

图 10-34 特殊功能寄存器

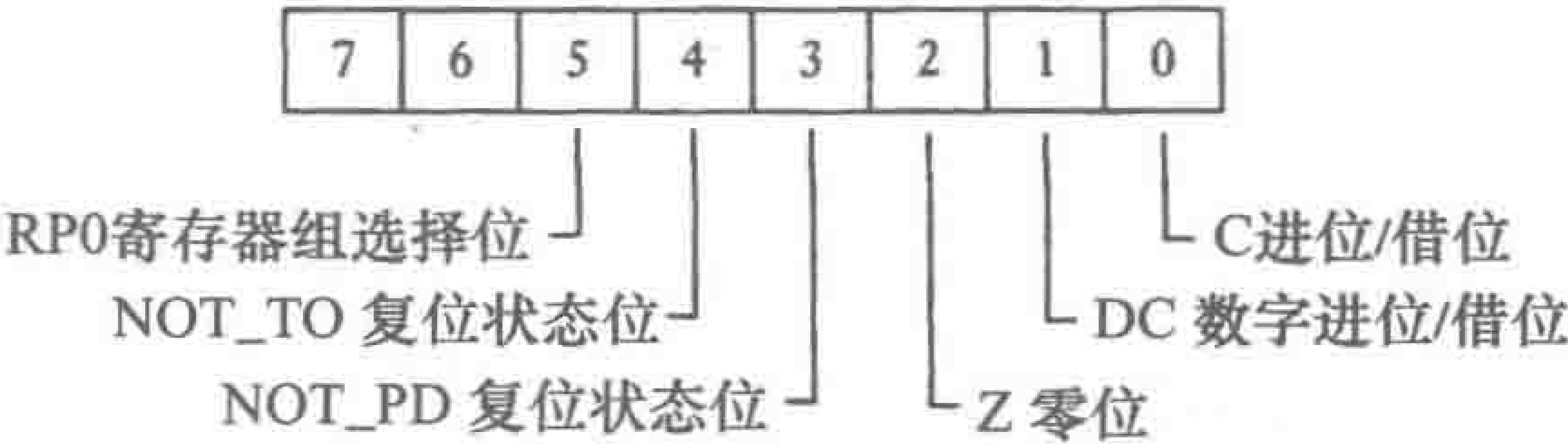


图 10-35 STATUS 寄存器

10.3.4 Atmel AVR 微控制器和 Arduino

Atmel AVR 有一系列的 8 位微控制器，它们使用了一种改进的哈佛结构，将程序与数据存储在分开的存储器位置。Arduino 是一种拥有补充元件的小型微控制器电路板，从而使微控制器在项目控制中的使用更便捷。Arduino UNO 修订版 3 是一款使用 Atmel 8 位微控制器 Atmega 328 的基本电路板。这款微处理器拥有存储器系统、输入/输出端口、计时器/计数器、脉冲宽度调制、ADC、中断器和串口通信模块。

Arduino 板可以从各零售商处预先装配购买。该板有通用串行总线（USB）接口，使其可

以直接连接电脑；也有一系列的连接插座，使其能够和电机、继电器等外部设备相连。该板可以通过连接外部电源来供电，如 9V 电池，或通过 USB 连接电脑供电。预先编程入板载微控制器芯片的是引导加载程序，它允许直接向微控制器的存储器上传程序。基本电路板由配件板来补充，配件板又称为扩展板，如 LCD（液晶）显示板、电动机板和以太网板。它们可以插在 Arduino 基本电路板的上面，然后插到排针里面。多个扩展板可以堆叠起来使用。

该板是**开源的**。这意味着任何人都可以制作兼容 Arduino 的板子。你可以找到新手工具箱，其中包含 Arduino 板、USB 线（用来进行电脑编程）、面包板（用线来装配外部电路），以及其他常用元件，包括电阻、光敏电阻电位器、电容器、按钮、温度传感器、LCD 字符显示屏、发光二极管（LED）、直流电动机、H 桥电动机驱动器、光耦合器、晶体管和二极管。

图 10-36 给出了 Arduino UNO 修订版 3 电路板的基本特征。

1）*在电源连接部分的复位管脚*。可以用来复位微处理器，从而重启程序。为了完成复位，这个管脚要瞬间被设成低电平，即与 0V 相连。该操作可通过使用复位开关来完成启动。

2）*电源连接器中的其他管脚*。可以提供不同的电压，即 3.5V、5V、GND 和 9V。

3）*模拟信号输入*。标记为 A0 到 A5，可以用来检测电压信号。

4）*数字信号连接*。标记为 D0 到 D13，可以被用作输入或者输出。头两个连接端子也被标记为 RX 和 TX，用于通信中的接收和发送。

5）*USB 连接器*。用来连接电脑。

6）*串行编程连接器*。该连接器为没有 USB 端口时进行 Arduino 编程提供了手段。

7）*LED*。该板装配有 3 个 LED。一个是用来指示串行发送（TX），一个用来指示串行接收（RX），另外一个 LED 可以在项目里面使用。

8）*供电连接器*。该板可以通过底部左边角上的连接器和一个外部电源相连。然而，在和电脑连接的时候，该板可通过 USB 接口供电。

9）*微处理器*。ATmega328 微控制器通过预烧引导加载程序来允许上载新程序，这样就可以不使用外部硬件编程器。

在购买的时候，Arduino 板通常有一个简单的预装程序，它的功能是让板上的 LED 闪烁。只需将板子通电，就可以运行该程序，你可以把板子连到电脑的 USB 端口来实现供电。于是，LED 应该闪烁，即证明该板正在工作。为了安装新软件（又称作“Arduino 板的 sketch”），你需要在电脑上安装 Arduino 软件并加载 USB 驱动程序。安装指南和软件见 Arduino 网站（www.arduino.cc）。一旦安装完成，即可向 Arduino 板上载程序。

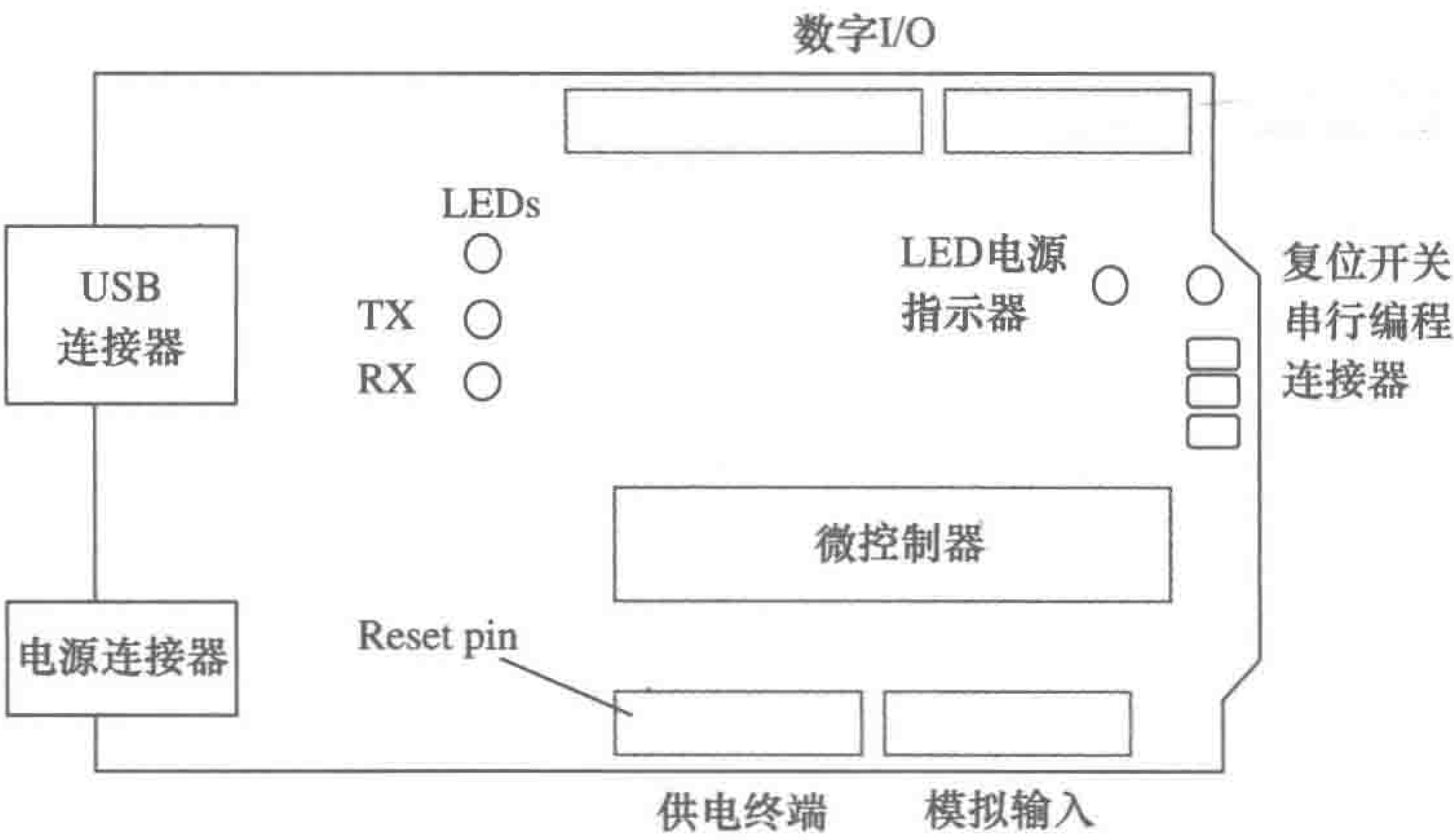


图 10-36 Arduino UNO 修订版 3 电路板的基本元件

10.3.5　选择微控制器

268
~
271

在选择微控制器时应考虑以下因素：

1. 输入/输出引脚的数量

对于给定的任务，我们需要多少个输入/输出引脚？

2. 接口的需求

我们需要什么样的接口？比如，是否需要 PWM？许多微控制器有 PWM 输出，比如 PIC17C42 就有两个。

3. 存储器需求

对于给定的任务，我们需要多大容量的存储器？

4. 中断数量的需求

有多少事件将需要中断？

5. 处理速度的需求

微处理器执行指令是需要时间的（见 11.2.2 节），这个时间由处理器的时钟决定。

为了说明不同的微控制器，表 10-1 列出了 Intel 8051 系列的详细成员，表 10-2 列出了 PIC16Cxx 系列的芯片，表 10-3 列出了 M68HC11 系列的芯片。

表 10-1　Intel 8051 系列芯片

	ROM	EPROM	RAM	定时器	I/O 端口	中断
8031AH	0	0	128	2	4	5
8051AH	4K	0	128	2	4	5
8052AH	8K	0	256	3	4	6
8751AH	0	4K	128	2	4	5

表 10-2　PIC16C 系列芯片

	I/O	EPROM	RAM	ADC 通道	USART	CCP 模块
PIC16C62A	22	2K	128	0	0	1
PIC16C63	22	4K	192	0	1	2
PIC16C64A	22	2K	128	0	0	1
PIC16C65A	33	4K	192	0	1	2
PIC16C72	22	4K	128	5	0	1
PIC16C73A	22	4K	192	5	1	2
PIC16C74A	33	4K	192	8	1	3

表 10-3　M68HC11 系列芯片

	ROM	EPROM	RAM	ADC	定时器	PWM	I/O	串口	外部时钟（MHz）
68HC11AO	0	0	256	8 通道，8 位	(1)	0	22	SCI，SPI	2
68HC11A1	0	512	256	8 通道，8 位	(1)	0	22	SCI，SPI	2
68HC11A7	8K	0	256	8 通道，8 位	(1)	0	38	SCI，SPI	3
68HC11A8	8K	512	256	8 通道，8 位	(1)	0	38	SCI，SPI	3
68HC11C0	0	512	256	4 通道，4 位	(2)	2 通道，8 位	36	SCI，SPI	2
668HC11D0	0	0	192	无	(2)	0	14	SCI，SPI	2

注：1. 定时器：(1) 包含 3 个输入捕获、5 个输出比较、实时中断、看门狗定时器、脉冲累加器；(2) 包含 3 或 4 个输入捕获、4 或 5 个输出比较、实时中断、看门狗定时器、脉冲累加器。
2. 串口：SCI 是异步串行通信接口，SPI 是同步串行外设接口。

10.4　应用

接下来的两个例子讲述的是如何应用微控制器。在第 24 章中会介绍更多的案例。

10.4.1　温度测量系统

作为微控制器应用的简单例子，图 10-37 给出了一个应用 MC68HC11 的温度测量系统的主要组成部分。温度传感器提供一个与温度成比例的电压信号（比如热晶体管 LM35，见 2.9.4 节）。温度传感器的输出和微控制器的 ADC 输入线相连。通过编程让微控制器把温度转换为 BCD 码输出，该输出可用于在一个 2 位 7 段数码管上显示。但是，因为温度会浮动，所以我们必须使用一个存储寄存器，让显示器有足够的时间读取数据。该存储寄存器，74HCT273 是一个 8 路 D 触发器，它在微控制器时钟输入的下一正跳变边沿复位。 272

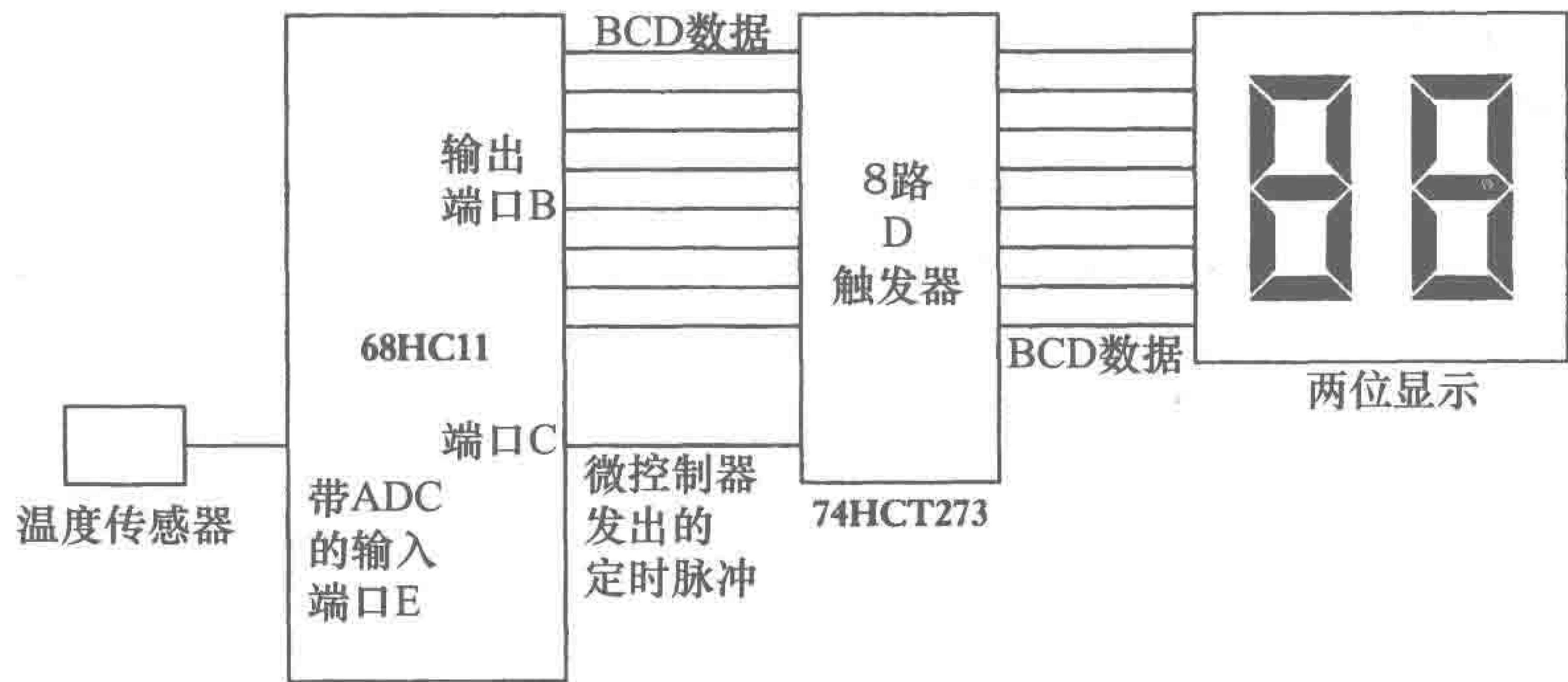


图 10-37　温度测量系统

10.4.2　家用洗衣机

图 10-38 展示了微控制器是如何运用在家用洗衣机中的。常用的微控制器为摩托罗拉 M68HC05B6，它比在本章前面提到的摩托罗拉 M68HC11 更简单和经济，并且广泛应用在低成本领域。

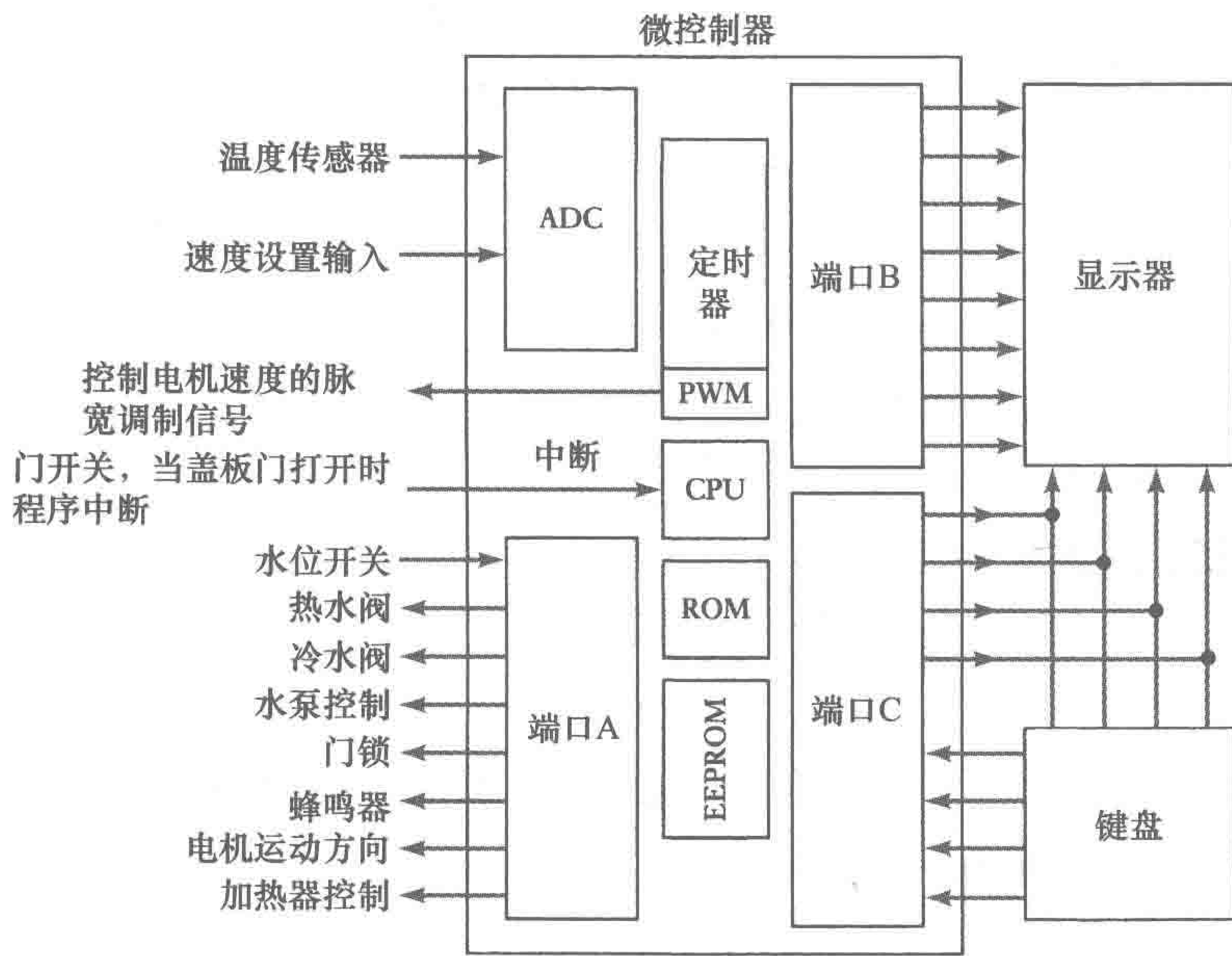


图 10-38　洗衣机

水温和电动机转速的传感器通过模数输入端口输入。端口 A 给各种用于控制机器的驱动器提供输出，并且它也是水位开关的输入。端口 B 负责将数据输出到显示屏。端口 C 为显示屏提供输出，并从键盘接收输入，该键盘用于给机器输入各种不同的程序选项。定时器的 PWM 部分提供 273

控制电动机转速的 PWM 信号。如果洗衣机的门被打开，则整个机器程序会被中断并停止。

10.5 编程

常用的程序开发方法遵循以下步骤：

1）定义问题、清晰地描述程序需要实现什么功能、输入和输出的需求、操作速度有什么限制、精度、存储器容量等。

2）定义算法。**算法**是指解决问题的方法的步骤序列。

3）对于少于几千条指令的系统来说，一个有效的方法是用流程图来表述算法。图 10-39a 是流程图准备过程中需要用到的标准符号。算法中的每一步用一个或多个这样的符号来表示，然后用线把这些符号连接起来表示程序的流程。图 10-39b 是一个具体的流程图，从程序开始，有一个操作 A，紧跟着可以转移到操作 B 或操作 C，这取决于条件判断中的值是“是”（YES）还是“否”（NO）。另一个有用的方法是**伪代码**。伪代码是指一种不正式地描述算法步骤的方法，之后它可以翻译成程序（见下一节）。

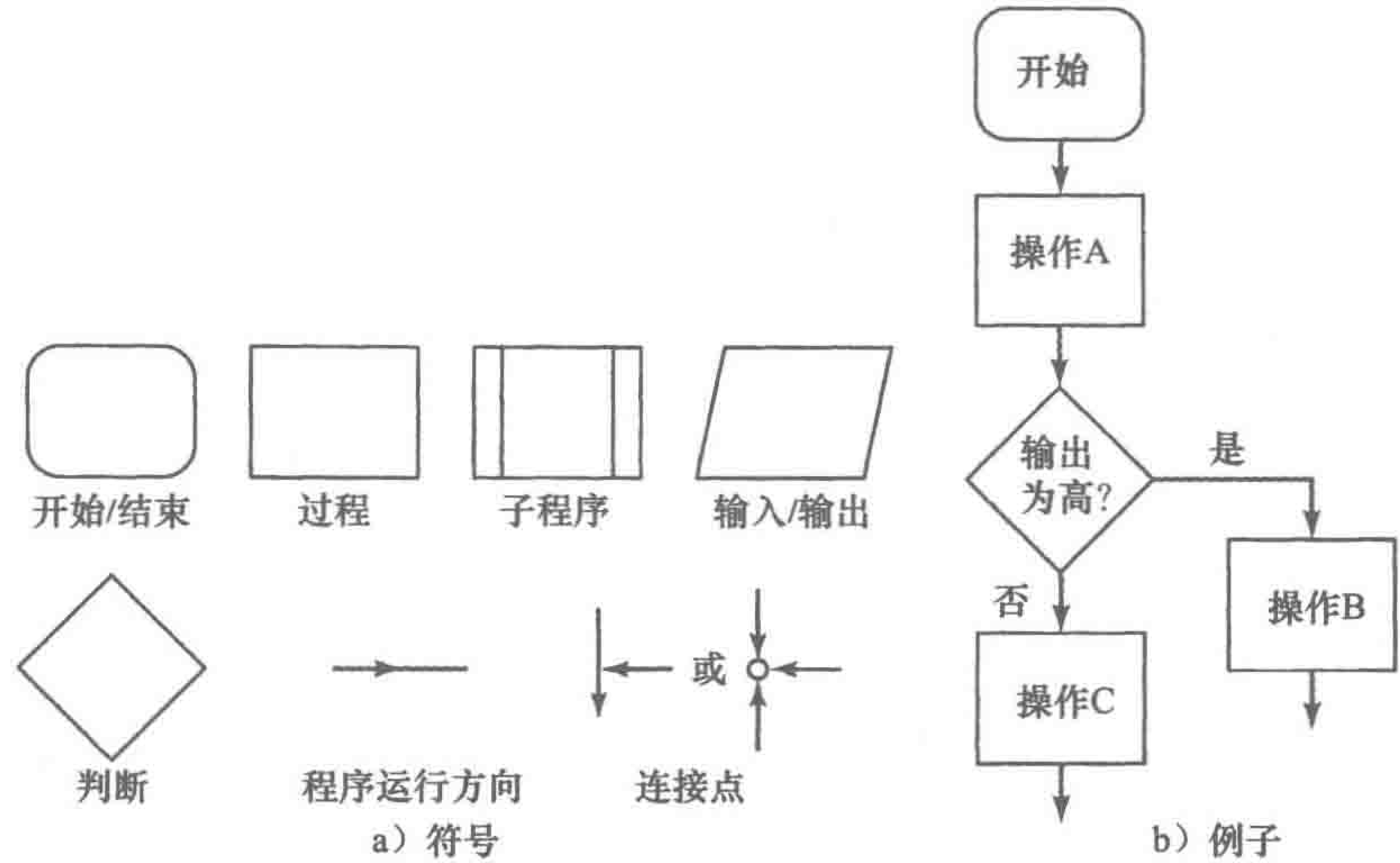


图 10-39 流程图 1

274

4）把流程图/算法翻译成微处理器可以执行的指令。这可以通过用某种程序语言来完成，比如汇编语言或 C 语言。然后把它人工地或由汇编程序自动地转换成微处理器可以接受的代码，即机器码。

5）测试和调试程序。程序中的错误称为**漏洞**（bug）。找到漏洞并去除它们的过程称为**调试**。

10.5.1 伪代码

伪代码和画流程图相似，它涉及程序编写，用判断语句 IF-THEN-ELSE 和循环语句 WHILE-DO 来实现功能或操作序列。

一个序列（见图 10-40a）可能写为：

```
BEGIN A
  ...
END A
  ...
BEGIN B
  ...
END B
```

一个判断可为：

```
IF X
THEN
  BEGIN A
    ...
  END A
ELSE
  BEGIN B
    ...
  END B
ENDIF X
```

图 10-40b 是流程图中的一个判断。循环可以写为：

```
WHILE X
DO
  BEGIN A
  ...
  END A
  BEGIN B
  ...
  END B
ENDO WHILE X
```

图 10-40c 用流程图来表示 WHILE-DO 语句。类似的程序可如下所示：

```
BEGIN PROGRAM
  BEGIN A
    IF X
      BEGIN B
      END B
    ELSE
      BEGIN C
      END C
    ENDIF X
  END A
  BEGIN D
    IF Z
      BEGIN E
      END E
    ENDIF Z
  END D
```

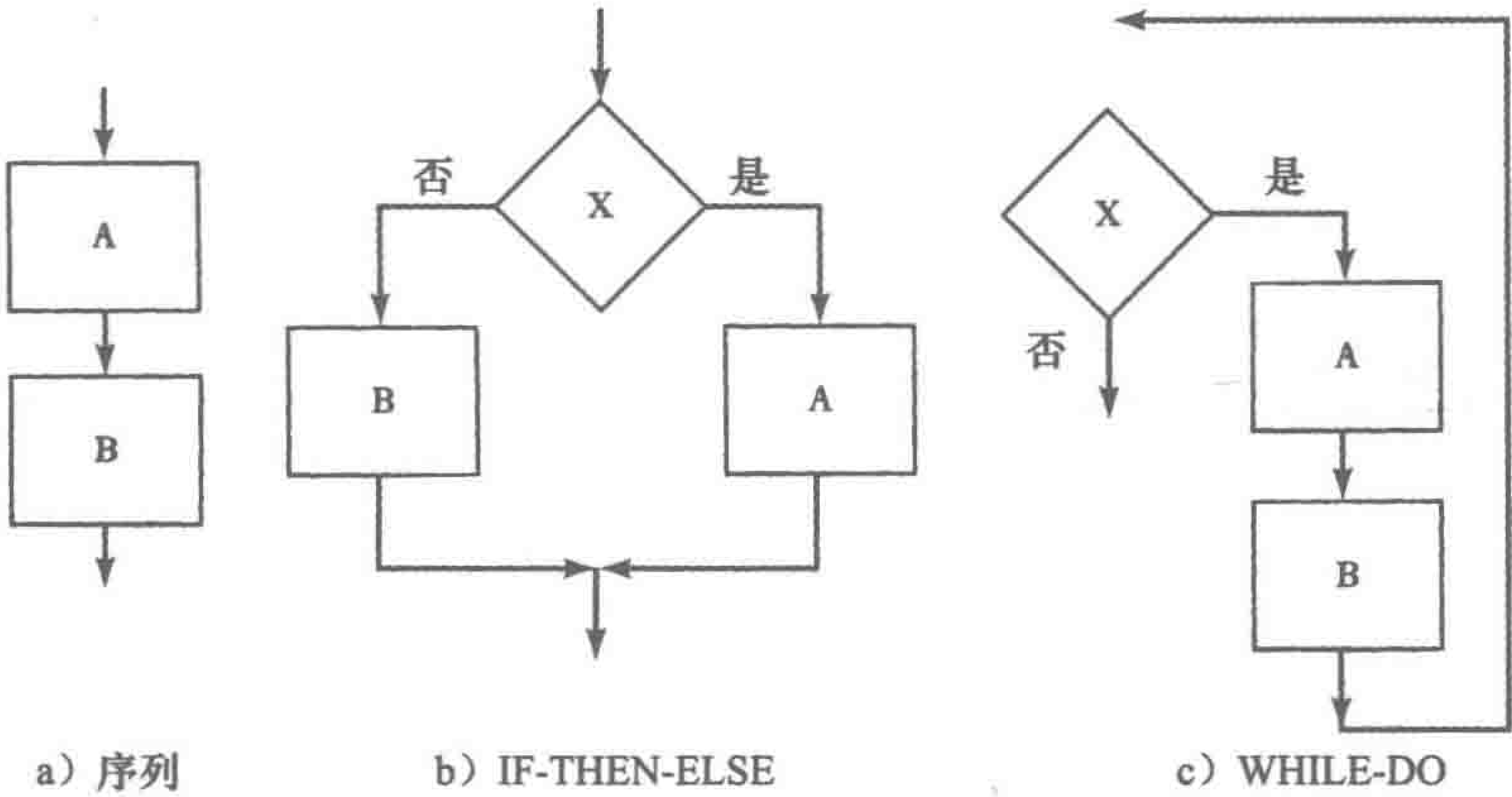


图 10-40　流程图 2

第 11 章将讲述如何用汇编语言编写程序，第 12 章则讲述如何用 C 语言编写程序。

本章小结

基本上，涉及**微处理器**的系统包括以下三个部分：中央处理单元（CPU）、输入/输出接口

和存储器。在微处理器中，数字信号通过**总线**并行传输，而不是串行传输。

微控制器把带有存储器的微处理器、输入/输出接口和定时器等其他外部设备集成到一个芯片中。

算法是用来定义解决一个问题的方法的步骤序列。流程图和伪代码是描述这些步骤的两种方法。

习题

10.1　对于一个微处理器，解释：(a) 累加器；(b) 状态；(c) 存储地址；(d) 程序计数器寄存器的作用。

10.2　一个微处理器用 8 根地址线访问存储器。它最多可以对多少个存储位置进行编址？

10.3　一个存储器芯片有 8 根数据线和 16 根地址线。它的容量是多少？

10.4　微控制器和微处理器的区别是什么？

10.5　画出一个基本微控制器的框图，并解释每个子系统的功能。

10.6　M68HC11 的哪个端口是用做 (a) ADC；(b) 双向端口；(c) 串行输入/输出；(d) 8 位输出端口？

10.7　M68HC11A7 一共有多少字节的存储空间来存放数据？

10.8　对于摩托罗拉 M68HC11，端口 C 是双向的。它是怎么被设置成 (a) 输入；(b) 输出的？

10.9　摩托罗拉 M68HC11 可以在单芯片模式下或者扩展模式下运行。为什么要有这些模式？

10.10　Intel 8051 的 ALE 引脚有什么作用？

10.11　要复位 Intel 8051 微控制器需要什么输入？

10.12　用伪代码来表示：

(a) 如果 A 是 yes 则 B，否则 C。

(b) 当 A 是 yes 时，执行 B。

275～277

第11章 汇编语言

本章目的

通过本章的学习，读者将能够：

- 使用汇编语言编写数据传送、算术运算、逻辑运算、跳转、转移、子程序、延时以及查表等程序的方法。

11.1 语言

软件（software）是用于告诉微型处理器或微型控制器做什么的**指令**。微处理器可以识别的指令集合是**指令集**。指令集因不同的微处理器而不同。一系列为了执行某一任务的指令叫作**程序**。

微处理器以二进制代码工作。二进制代码写成的指令称为**机器码**。用这种指令写程序是一个需要技巧且十分烦琐的任务。因为机器码程序是由一系列的0和1构成的，指令很难通过其组合来理解，因此极易出错。一种代替方法是用简单易懂的速记码来代替0和1的组合。例如在“原数值基础上增加一个量”这个功能可以简单地记为ADDA。这种速记码是一种**助记码**，是一种方便记忆的编码。**汇编语言**就是这种编码。用助记码写程序更加简单，因为它们是指令操作的缩略版本。除此之外，这种描述程序的指令可更好地理解，比二进制机器码更能减少错误。汇编程序还可以转换为微处理器可以处理的机器码。这种转换可以通过手动对照厂家给出的数据手册中的助记符-二进制码表来完成。当然，这种转换还可用程序来完成，这种程序称为**汇编器程序**。

高级语言，例如BASIC、C、FORTRAN和PASCAL，是一类描述操作所需的更简单、更易于理解的程序语言。这些语言必须被计算机程序转化为微处理器可以使用的机器码。本章介绍怎样使用汇编语言写程序，第12章将介绍如何用C语言写程序。

11.2 指令集

下面是微处理器经常用到的指令，我们把这样的整个指令表叫作指令集。附录D给出了3种常见微处理器的指令集。每个微处理器的指令集都不同。以下是一些常见的指令：

数据传输和转移

1. 加载（Load）

该指令读取指定存储器的内容并将数据复制到中央处理单元（CPU）的一个指定寄存器中，以摩托罗拉的微处理器为例，LDAA $0010指令所代表的意思是：

指令执行前	**指令执行后**
数据在0010单元中	数据在0010单元中
	数据从0010单元进入累加器A中

2. 存储（Store）

该指令把指定寄存器的内容复制到指定存储器单元，以摩托罗拉的微处理器为例，STA

$0011指令所代表的意思是：

指令执行前	指令执行后
数据在累加器 A 中	数据在累加器 A 中 数据复制到 0011 单元中

3. 移动（Move）

该指令用于把数据移动到一个寄存器中，或者从一个寄存器复制到另一个寄存器中，以外围接口控制器（PIC）和 Intel 微处理器为例，如 PIC 微控制器的 MOV R5，A 指令所代表的意思是：

指令执行前	指令执行后
数据在寄存器 A 中	数据在寄存器 A 中 数据复制到寄存器 R5 中

4. 清除（Clear）

该指令用于把所有位清零，以摩托罗拉微处理器为例，CLR A 用于清除累加器 A；以 PIC 微控制器为例，CLRF 06 用于清除寄存器 06。

算术运算

5. 加（Add）

该指令用于把一个数加到某个寄存器中的数据上，以 Intel 为例，ADD A，#10h 的含义是：

指令执行前	指令执行后
数据在累加器 A 中	累加器 A 加 10 hex

以摩托罗拉微处理器为例，ADD D，$0020 的含义是：

指令执行前	指令执行后
数据在累加器 D 中	累加器 D 加上 0020 单元的内容

279

或者一个寄存器的内容加上另一个寄存器的数据，以 Intel 为例，ADD A，@ R1 的含义是：

指令执行前	指令执行后
数据在累加器 A 中	累加器 A 加上 R1 所指的 RAM

以 PIC 微控制器为例，addwf 0C 的含义是：

指令执行前	指令执行后
数据在寄存器 0C 中	寄存器 0C 加单元 w 内的地址所指的内容

6. 减 1（Decrement）

该指令用于将特定单元内的数据减 1。我们用寄存器 3 作为特定地址，以 Intel 为例，DEC R3：

指令执行前	指令执行后
寄存器 R3 中的数据为 0011	寄存器 R3 中的数据变为 0010

7. 加 1（Increment）

该指令将特定单元中的数值加 1，例如 INC A 在摩托罗拉处理器中使累加器 A 中的数据增加 1；在 PIC 微控制器中，指令 incf 06 使寄存器 06 中的数据增加 1。

8. 比较（Compare）

该指令比较某一寄存器中的数据是大于、等于，还是小于指定寄存器中的数据。其结果出现在状态寄存器中的一个标志位（flag）上。

逻辑运算

9. 与（AND）

该指令把一个存储器中的数据与某一寄存器中的数据逐位执行逻辑与运算。以摩托罗拉处

理器为例，ANDA%1001 的含义是：

指令执行前	指令执行后
累加器 A 中的数据 0011	累加器 A 中的数据 0001
存储器中的数据 1001	

两个数据中只有最后一位全都是 1，与运算后在结果中只有最后一位是 1。以 PIC 为例，ANDLW 01 将二进制数 01 与 W 中的数相加，如果 W 中数的最后一位是 0，那么结果是 0。

10. 或（OR）

该指令把一个存储器中的数据和某一寄存器中的数据逐位执行逻辑或运算，以 Intel 处理器为例，ORL A，#3Fh 在寄存器 A 与十六进制（hex）数 3F 之间进行或运算。

11. 异或（EXCLUSIVE－OR）

该指令把一个存储器中的数据和某一寄存器中的数据逐位进行逻辑异或运算，以 PIC 微控制器为例，xorlw 81h（二进制 10000001）：

280

指令执行前	指令执行后
寄存器 w 中的数据 10001110	寄存器 w 中的数据 00001111

异或运算中同一位上两个数相同得 0，不同得 1。

12. 逻辑移位（logical shift）（向左或向右）

逻辑移位指令是通过在寄存器内的数据的一端移入一个 0，把数据位向左或向右移动一位。例如在逻辑右移中，0 被插入到最高位，而最低位被移到状态寄存器中的进位标志位中。摩托罗拉的右移指令为 LSRA，左移指令为 LSLA。右移指令举例：

指令执行前	指令执行后
累加器中的数据 0011	累加器中的数据 0001
	状态寄存器进位标志位为 1

13. 算术移位（arithmetic shift）（向左或向右）

算术移位指令保留寄存器中数据左端的符号位，并把数据位向左或向右移动一位。以摩托罗拉为例，算术右移指令 ASRA 的含义是：

指令执行前	指令执行后
累加器中的数据 1011	累加器中的数据 1001
	状态寄存器进位标志位为 1

14. 循环移位（Rotate）（向左或向右）

循环移位指令包括把寄存器中的数据向左或右移动一位，多出来的位被写回到另一端。以 Intel 的循环右移 RR A 为例：

指令执行前	指令执行后
累加器中的数据 0011	累加器中的数据 1001

程序控制

15. 跳转或转移（Jump or branch）

该指令改变程序执行的顺序。通常程序计数器使程序按严格的数字顺序执行。而跳转指令可使程序计数器跳转到程序的其他指定地址执行（见图 11-1a）。无条件跳转在不需程序测试的条件发生。以 Intel 为例，LJMP POINTA 使程序跳转至标记为 POINTA 行；而摩托罗拉的类似指令为 JMP POINTA，PIC 的类似指令为 GOTO POINTA。条件跳转是在特定条件满足后发生的（见图 11-1b）。如果累加器中数据的任何一位不为 0，则 Intel 的 JNZ POINTA 指令使程序跳转到 POINTA 行，否则就进入下一行。JZ POINTA 是累加器中所有位都是 0 时执行跳转的指令。

PIC 中包含两行代码的一种条件跳转指令为 BTFSC 05，1，执行“检测并跳过”，即测试寄存器 5 中的第 1 位，如果是 0 则跳过下一程序行，是 1 则顺序执行。另一种无条件跳转指令为 GOTO POINTA。在摩托罗拉的转移指令是一种条件跳转指令，用来决定某一条件发生时程序运行的分支。例如，摩托罗拉用比较指令 BEQ，如果比较结果等于 0，则执行分支程序。比较指令 BGE，比较结果大于等于 0 时启动分支程序。比较指令 BLE，比较结果小于等于 0 时启动分支程序。

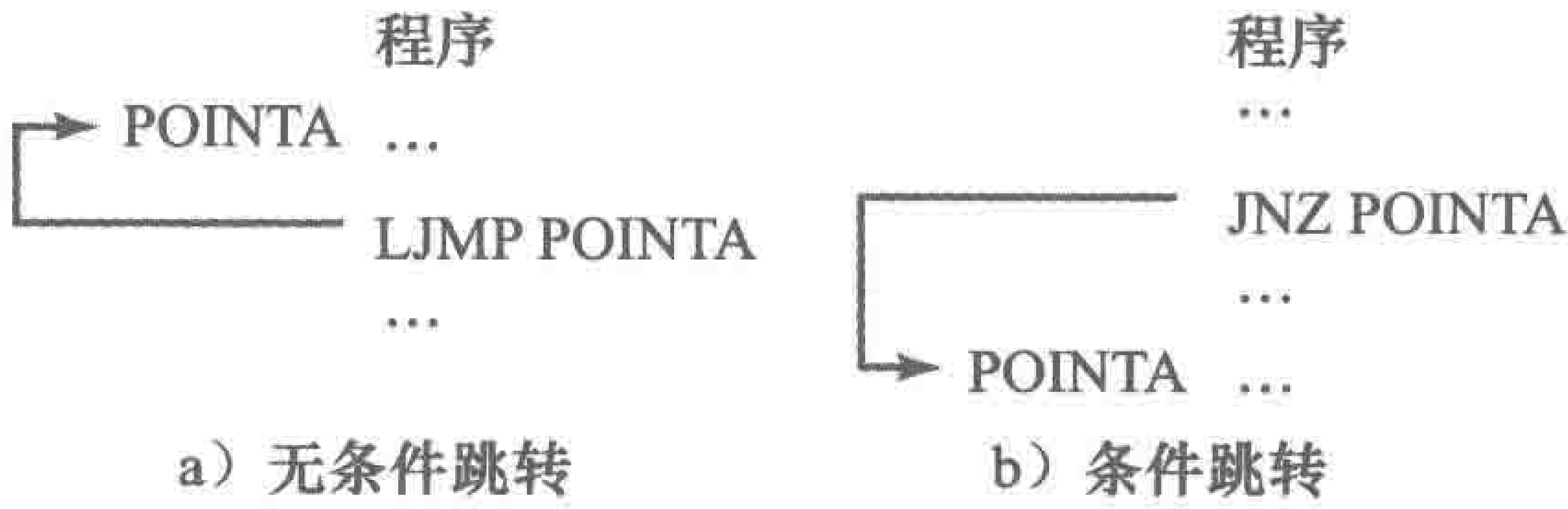


图 11-1　跳转

16. 中断/停止（Halt/stop）

该指令停止微处理器的所有动作。

数字数据可能是二进制、八进制、十六进制或十进制数。如果没有任何提示，那么汇编器通常认为数据是十进制的。摩托罗拉的数字前加前缀#；二进制加前缀% 或后缀 B；八进制加前缀@ 或后缀 O；十六进制加前缀 $ 或后缀 H；十进制不需要加字母或提示符。以 Intel 为例，数值必须加前缀#来表明是数值，而后缀二进制加 B，八进制加 O 或 Q，十六进制加 H 或 h，十进制加 D 或不加。PIC 微处理器的头文件有 R = DEC，表示无特别指明的数字是十进制数。二进制被引号括起并且前面加 B，十六进制前加 H。

11.2.1　寻址

当一个助记符，如 LDA 被用做一个指令时，它的后面将要跟着额外的信息来指定指令所需的数据来源和目的地。跟在指令后的数据称为**操作数**（operand）。

有很多不同的确定数据地址的方法，即寻址，因此程序以这些寻址方式使微处理器获得自己的指令或数据。不同的微处理器有不同的寻址方法。摩托罗拉 68HC11 有立即寻址、直接寻址、扩展寻址、变址寻址、固有寻址、相对寻址 6 种寻址方法；Intel 8051 有立即寻址、直接寻址、寄存器寻址、间接寻址和变址寻址 5 种寻址方法；PIC 微处理器有立即寻址、直接寻址和允许变址寻址的间接寻址 3 种。以下是常用的寻址模式：

1. 立即寻址

立即寻址方式所提供的操作数紧跟在操作码的后面，用于加载预定值到寄存器或存储器中。例如，对于摩托罗拉指令码，LDA B # $25 用于把 25 加载到累加器 B 中。#代表立即寻址模式和数值，$ 代表数字是十六进制数。对于 Intel 指令码，MOV A，#25H 把 25H 加载到累加器 A 中。#代表一个数字，H 代表十六进制。对于 PIC 指令码，movlw H '25' 把 25H 加载到工作寄存器 w 中，H 代表十六进制数。

281～282

2. 直接寻址、绝对寻址、扩展寻址和零页寻址

用直接寻址、绝对寻址、扩展寻址和零页寻址，操作码后的字节直接给出了将要在指令中使用的数据地址。当只有 8 位地址时，用**直接寻址**；当有 16 位宽的地址时，用**扩展寻址**。例如，对于摩托罗拉指令码，LDAA　$25 表示在累加器中加载 0025 存储器地址的数据。对于 Intel 指令码，为了完成同样的操作，可用直接寻址指令 MOV A，20H 把地址 20 中的数据复制到累加器 A 中。对于 PIC 指令码，指令 Movwf Reg1 把 Reg1 中的内容复制到工作寄存器中，Reg1 的地址已被事先定义过。

3. 隐含寻址或固有寻址

在隐含寻址或固有寻址模式下，地址被隐含在指令的操作码中。例如，对于摩托罗拉和 Intel 指令码，CLR A 表示清空累加器 A。PIC 指令 clrw 表示清空工作寄存器。

4. 寄存器

在这种寄存器寻址模式下，操作数被指定为一个内部寄存器的内容。例如，对于 Intel 指令码 ADD R7,A，把累加器 A 中的数据加到寄存器 R7 中。

5. 间接寻址

这种间接寻址寻址方式表示数据在指令给定地址的存储器单元中找到。例如，在 PIC 系统中使用的寄存器是 INDF 和 FSR。首先，地址被写入 FSR 寄存器中，之后它作为一个地址指针工作。随后应用 movf INDF，w 指令对 INDF 进行直接访问，将在工作寄存器 w 中加载指针 FSR 的内容作为地址的数据。

6. 变址寻址

变址寻址将数据的地址保存在一个变址寄存器中。指令的第一个字节包括操作码，第二个字节包含地址偏移量。变址寄存器的内容与指令中给出的地址偏移量相加，从而得到一个操作数的地址。摩托罗拉的变址指令可能为 LDA A $ FF,X。变址寄存器的内容加上 FF 得到的地址的数据加载到累加器 A。另一个例子为 STA A $05,X，它表示把累加器 A 中的数据存储到地址为 05 加上变址寄存器所得的地址中。

7. 相对寻址

相对寻址寻址方式和转移指令一起使用。操作码后的一个字节数据叫作相对地址。这表示如果转移条件满足，则地址偏移量必须加到程序指针上。例如，摩托罗拉的指令码 BEQ $F1 表示，如果 CPU 执行的操作结果数据等于0，则程序执行的下一地址再向前 F1。即 F1 的相对地址被加至下一条指令的地址上得到转移的目的地址。

作为说明，表 11-1 给出了摩托罗拉系统的寻址模式的一些指令。

283

表 11-1 寻址的例子

寻址模式	指令	
立即寻址	LDA A # $F0	累加器 A 赋值为 F0
直接寻址	LDA A $50	将累加器 A 赋值为地址是 0050 的寄存器中的值
扩展寻址	LDA A $0F01	将累加器 A 赋值为地址是 0F01 的寄存器中的值
变址寻址	LDA A $CF, X	将累加器 A 赋值为变址寄存器加上 CF 所确定地址的寄存器中的值
固有寻址	CLR A	清除寄存器 A
扩展寻址	CLR $2020	清除地址为 2020 寄存器的值，即在 2020 存储所有的 0
变址寻址	CLR $10, X	清除地址为变址寄存器加上 10 所确定地址的寄存器的值，即在这个地址中存储所有的 0

11.2.2 数据移动

以下是可以在厂商（6800）的指令集表格中找到的信息类型的例子：

		寻址模式					
		立即			直接		
操作	助记符	OP	~	#	OP	~	#
Add	ADDA	8B	2	2	9B	3	2

~代表微处理器所需机器周期的个数，#代表所需的程序字节数。

这意味着在立即寻址模式下，处理器的加法操作由助记符 ADDA 代表。当应用立即寻址方式时，对应的机器码是 8B，需要执行两个机器周期。这一操作将在程序中需要两字节。**操作码**（op-code 或 operation code）指明微处理器将要执行的工作，并以十六进制形式表示。1 字节

是由 8 位二进制数组成，被微处理器识别为一个字符。所以立即寻址需要两字符。在直接寻址模式下，对应的机器码是 9B，需要 3 个机器周期，占用两个程序字节。

为了说明数据信息如何在处理器和存储器之间传递，考虑以下任务。对于下面的例子，为方便，一个新程序在 RAM 中的地址从 0010 开始。为了使用直接寻址，程序地址必须处于零页，即在 0000 ~ 00FF 之间。这些例子都是基于 M6800 微处理器的指令集。

任务 1：把累加器 A 中的所有数据变为 0。

存储器地址	**操作码**	
0010	8F	CLR A

284

因为 CLR A 在程序中只占 1 字节，所以下一个可用的地址是 0011，这是固有寻址模式。

任务 2：把数据 20 加到累加器 A 中。

存储器地址	**操作码**	
0010	8F 20	ADD A # $20

这里使用了立即寻址模式。因为在这种寻址模式下 ADD A 占用 2 字节，下一个可用地址是 0012。

任务 3：把地址为 00AF 的存储器中的数据加载进累加器 A 中。

存储器地址	**操作码**	
0010	B6 00AF	LDA A $00AF

这里使用了绝对寻址模式，因为在这种寻址模式下 LDA A 占用 3 个程序字节，下一个可用地址为 0013。

任务 4：把存储器地址 00AF 内的数据循环左移。

存储器地址	**操作码**	
0010	7900AF	ROL $00AF

这里使用了绝对寻址模式。因为此模式下 ROL 占用 3 个程序字节，下一个可用地址为 0013。

任务 5：把累加器 A 中数据存入存储器地址 0021 中。

存储器地址	**操作码**	
0010	D721	STA A $21

这里使用了直接寻址模式。因为此模式下 STA A 占用 2 个程序字节，下一个可用地址为 0012。

任务 6：如果前一个指令结果为零，向前转移 4 个地址。

存储器地址	**操作码**	
0010	2704	BEQ $04

这里使用了相对寻址模式。因为此模式下 BEQ 占用 2 个程序字节，下一个可用地址在结果不为 0 时，为 0012。如果结果是 0，则下一个地址是 0012 + 4 = 0016。

11.3　汇编语言程序

一个汇编语言程序是由一系列指令组成的，经汇编器（assembler）处理后将转换成机器码程序。用汇编语言写的程序是由一系列语句组成的，一行一个语句。每一条语句可以由 4 个**字段**组成，格式如下：

标号（Label）　操作码（Op-code）　操作数（Operand）　注释（Comment）

一个特殊符号用于表示语句的结束或开始，具体符号取决于不同的微处理器的机器码汇编器。摩托罗拉6800中使用空格。Intel 8080中在标号后使用冒号，操作码后是空格，在地址字段中的条目之间用逗号，在注释前用分号。在一般情况下，分号用来分开注释和操作数。

285

标号是指访问存储器特定地址的名字。标号可以由字母、数字和一些其他符号组成。摩托罗拉6800中，标号只能是1~6个字符，第一个字符必须是字母，并且不能只是A、B、X单个字符，因为这些字符是为累加器和变址寄存器预留的。Intel 8080中，标号是5个字符，第一个字符是字母、@或?。标号不应使用任何为寄存器、指令代码或伪指令（见本节后面的部分）预留的字符。程序中的所有标号必须是唯一的。如果没有标号，则必须用空格在该字段占位。对于摩托罗拉6800，在标号字段的星号（*）代表整条语句都是注释，该注释使程序更加清楚。在机器码程序的编译过程中，注释部分将会被汇编器忽略。

操作码说明了数据是怎样处理的，并有它自己的助记符，例如LDA A。这是唯一永远不能为空的字段。此外，这个字段可以包含汇编器的伪指令，这些指令又称为伪操作，因为它们出现在操作码字段，但并未翻译成机器码指令。但它们可定义符号、给程序和数据分配一定的存储器空间、生成固定表格和数据，以及指明程序结束等。常用的汇编伪指令有：

设定程序计数器

ORG	该指令定义紧跟在后面程序部分的存储器开始地址。一个程序可能有多个开始点

定义符号

EQU，SET，DEF	给一个数赋值、设置、定义一个符号或一个表达式

保留存储器单元

RMB，RES	保留存储器字节/空间

在存储器内定义常量

FCB	定义常量字节
FCC	定义常量字符串
FDB	定义双字节常量
BSZ	清零存储器

操作数字段内的信息取决于它前面的助记符和使用的寻址模式。它给出了操作码指定的被操作数据的地址。因此常常称为**地址字段**。如果操作码给定的操作不需要任何数据或地址，那么此字段可以为空。这一字段的数值可以是十六进制、十进制、八进制或二进制。汇编器默认数据是十进制的，除非另有指定。对于摩托罗拉6800指令系统，十六进制数加前缀$或后缀H；八进制数加前缀@或后缀O、Q；二进制数加前缀%或后缀B。对于Intel 8080指令系统，十六进制数加后缀H，八进制数加后缀O或Q，二进制数加后缀B。十六进制数必须以十进制数0~9开始，以避免与名称混淆。对于摩托罗拉6800指令系统，立即寻址模式可通过在操作数前加前缀#表示，变址寻址模式的操作数后加后缀X。直接寻址和扩展寻址不加任何符号。如果地址在零页，即小于等于FF，则汇编器自动分配直接寻址模式。如果地址大于FF，则汇编器分配扩展寻址模式。

286

注释字段可以不填写，或填写能使读者更容易理解的程序注释。在汇编器生成机器码时，注释字段被忽略。

11.3.1 汇编语言程序的范例

以下例子将说明如何开发简单的程序。

问题：两个存在于不同存储器地址的 8 位数相加，并且把结果存入存储器。

算法如下：

1）开始。

2）在累加器中加载第一个数。累加器是专门存放算术或逻辑运算的操作数和运算结果的工作寄存器，它像一个记事本一样，先在其上进行计算，然后结果再被转移到其他地方。所以我们在算术运算执行前，将数据复制到累加器中。对于 PIC 指令系统，使用工作寄存器（w）。

3）加上第二个数。

4）把和存入指定的存储器地址。

5）停止。

图 11-2 用流程图给出了这些步骤。



图 11-2　两个数相加的流程图

以下是为三种不同的微处理器写的程序。每一个程序中，第一列给出了标号，第二列是操作码，第三列是操作数，第四列是注释。所有注释都前加分号。

```
M68HC11 程序
      ;两数相加
NUM1  EQU   $00    ;第一个数的地址
NUM2  EQU   $01    ;第二个数的地址
SUM   EQU   $02    ;和的地址
      ORG   $C000  ;用户 RAM 的初始地址
START LDAA  $NUM1  ;在寄存器 A 中加载第一个数
      ADDA  $NUM2  ;在寄存器 A 中加入第二个数
      STAA  SUM    ;把和存入 $02
      END
```

程序中第一行指定了第一个要加的数的地址。第二行指定了要加到第一个数上的数的地址。第三行指定了和存放的位置。第四行指定程序应该开始的存储器地址。使用标号意味着与数据有关的操作数只用标号，而不用指定地址。

287

在 Intel 8051 微处理器中，同样的程序如下：

```
8051 程序
      ;两数相加
NUM1  EQU  20H    ;第一个数的地址
NUM2  EQU  21H    ;第二个数的地址
SUM   EQU  22H    ;和的地址
      ORG  8000H  ;用户 RAM 的初始地址
START MOV  A,Num1 ;在寄存器 A 中加载第一个数
      ADD  A,Num2 ;在寄存器 A 中加入第二个数
MOV   SUM  ,A     ;把和存入 22H
      END
```

在 PIC 微控制器中，同样的程序如下：

```
PIC  程序
        ;两数相加
NUM1    equu     H'20'    ;第一个数的地址
NUM2    equ      H'21'    ;第二个数的地址
SUM     equ      H'22'    ;和的地址
        org      H'000'   ;用户 RAM 的初始地址
START   movlw    Num1     ;在寄存器 w 中加载第一个数
```

```
addlw    Num2     ;在寄存器 w 中加入第二个数
movwf    SUM      ;把和存入地址 22H
End
```

很多程序中，可能要求一个任务被顺序执行多次。这种情况下，程序可以设计成多次执行的相同部分。这叫作**循环**（looping）。一个循环是被重复执行多次的程序段。图 11-3 说明了一个循环的流程图。在该循环中，程序在继续向下执行之前，某些操作要执行多次。只有这些操作次数完成之后，程序才能继续进行下去。以下问题就针对了这样的循环。

问题：将来自 10 个不同地址的数据相加（它们可能是来自被分别采样的 10 个不同传感器的输入结果）。

算法如下：

1)开始。
2)定义计数值为 10。
3)指向最下面的地址。
4)加上最下面的地址号。
5)计数值减 1。
6)地址指针加 1。
7)计数值是否为 0? 如果不是，跳转至 4;如果是，继续。
8)存储和。
9)结束。

图 11-4 为该算法的流程图。



图 11-3　循环

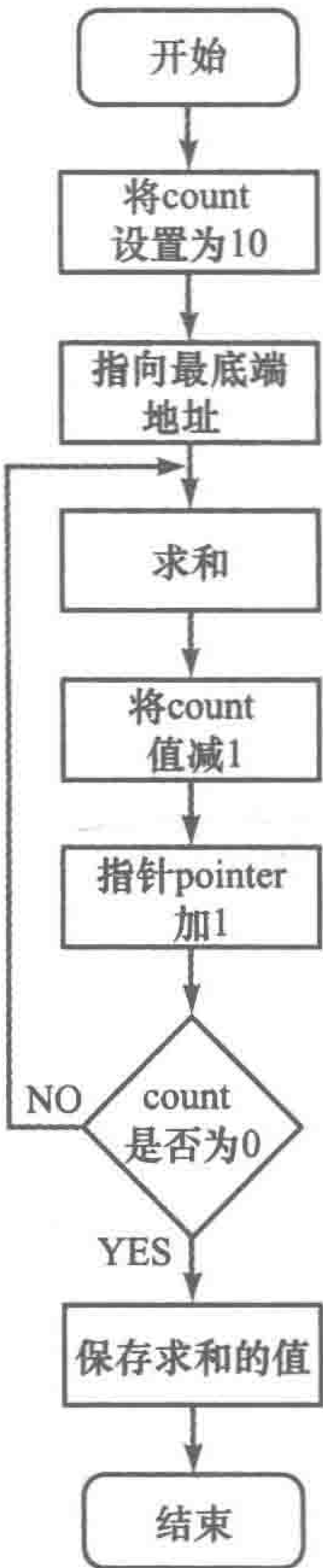


图 11-4　累加 10 个数的流程图

程序为：

```
COUNT  EQU    $0010
POINT  EQU    $0020
RESULT EQU    $0050
       ORG    $0001
       LDA B  COUNT    ;加载计数值
       LDX    POINT    ;初始化变址寄存器
SUM    ADD A  X        ;加被加数
       INX             ;变址寄存器加 1
       DEC B           ;累加器 B 减 1
       BNE    SUM      ;不为 0 跳转到 SUM
       STA A  RESULT   ;存储
       WAI             ;停止程序
```

计数值 10 被加载到累加器 B 中。变址寄存器给出了被加数的初始地址。求和的第一步是把变址寄存器给出的存储器地址中的内容和累加器 A 中的内容相加，假设累加器 A 已经初始化为 0（CLR A 可以用于最初清零）。INX 指令使变址寄存器加 1，使下一个被寻的地址为 0021。DEC B 使累加器 B 减 1，表明还剩下 9 次。然后，BNE 指令判断如果 B 不为 0，则使程序转移到标号 SUM，即 Z 标志位为 0 值。程序循环，直到累加器 B 为 0。

问题：在一系列数字中找出最大数（该方法可用于确定一系列温度传感器的最大温度值）。

算法如下：

1）清空存放答案的地址。
2）列出起始地址。
3）从起始地址中加载数据。
4）与存放答案的地址中的数比较大小。
5）如果答案的地址中的数更大，存储原答案值。
6）如果加载数大，存储加载数。
7）起始地址加 1。
8）如果地址不是最后一个，跳转至 3。
9）停止。

图 11-5 为流程图。程序为：

```
FIRST  EQU    $0030
LAST   EQU    $0040
ANSW   EQU    $0041
       ORG    $0000
       CLR    ANSW     ;清空存放答案的地址
       LDX    FIRST    ;加载起始地址
NUM    LDA A  $30,X    ;加载数
       CMP A  ANSW     ;比较加载数和存放答案的地址里的数大小
       BLS    NEXT     ;如果小于等于，跳转至 NEXT
       STA A  ANSW     ;存储答案值
NEXT   INX             ;增加变址地址
       CPX    LAST     ;比较变址地址和 LAST
       BNE    NUM      ;如果不等于 0 跳转
       WAI             ;停止
```

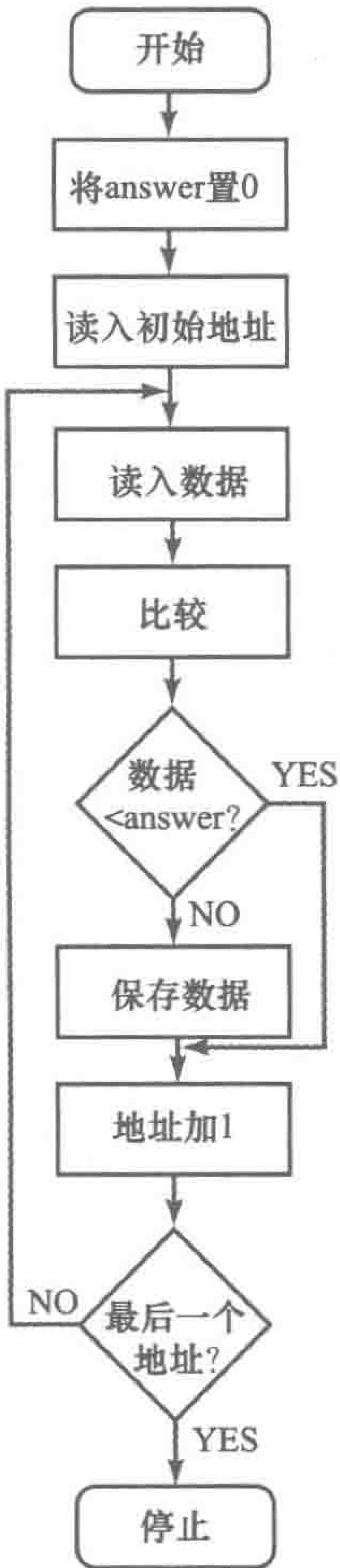


图 11-5　求最大值流程图

首先，程序清空存放答案的地址。然后，加载第一个地址，地址中的数放入累加器 A 中。

LDA A $30, X 指令表示加载变址寄存器加上30后的地址中的数到累加器A中。比较这个数和存放答案的地址中的数哪个大，如果这个数大，则保留此数；否则，跳转到循环处理下一个数。

11.4 子程序

在程序中可能还常常需要一块子程序出现几次。例如，可能需要一块程序产生延时。这时可以在主程序中复制几遍子程序。但是，这是对存储器使用的浪费。另一种方法，我们可在存储器内只有一个子程序，每当需要时，就跳转或转移到子程序。然而，这就提出了当子程序完成之后，在主程序的何处继续执行下去的问题。这就需要一个返回到主程序，并且继续从离开主程序执行子程序的点执行下去的机制。为此，我们需要存储跳转到子程序时的程序计数器的值，这样当子程序完成后，这一数值可以被重新载入程序计数器。大多数微处理器都提供了如下两个指令，使子程序这样来执行：

1）JSR（跳转至子程序），或 CALL，可以调用子程序。

2）RTS（从子程序返回），或 RET（返回），作为子程序的最后一条指令，返回到调用程序的正确位置。

子程序在一个程序的不同地址都可能被调用。所以，我们必须用一种后入先出（Last-In-First-Out，LIFO）的存储方法来保存程序计数器的内容。该寄存器涉及一个**栈**。就像一叠盘子，最后一个盘子总是被加在一叠盘子的最上面，第一个被拿走的盘子也是在最上面的，因此我们说最后的盘子被加到了栈上。这一栈可能是微处理器中的一块寄存器组，或者更常见的是用 RAM 的一部分。微处理器中有一个特殊寄存器叫作栈指针寄存器，该寄存器用来指向下一个用于栈的 RAM 区的自由地址。

除了在应用子程序时自动地使用栈外，程序员还可以编写使用栈的程序来暂存数据。两个有关的命令为：

1）PUSH，使指定寄存器中的数据存储到栈的下一个地址。

2）PULL，或 POP，从栈顶取回数据，并转移到指定的寄存器。

289
~
290

例如，在进入子程序前，某些寄存器中的数据必须被存储，在子程序调用后数据必须被恢复。对于摩托罗拉6800，可能的程序段为：

```
SAVE      PSH A  ;把累加器A推入栈
          PSH B  ;把累加器B推入栈
          TPA    ;把状态寄存器移入累加器A
          PSH A  ;把状态寄存器推入栈
;子程序
RESTORE   PUL A  ;从栈中返恢复状态码至累加器A
          TAP    ;从A中恢复状态码至状态寄存器
          PUL B  ;从栈中恢复累加器B
          PUL A  ;从栈中恢复累加器A
```

11.4.1 延时子程序

当微处理器有来自数模转换器等设备的输入时，常常需要**延时循环**。要求输入到转换器的信号开始转换后，在从转换器读出数据前，等待一段固定时间。在继续执行其余程序前，可通过一个循环使微处理器执行一系列的指令操作，从而达到延时的效果。简单的延时程序为：

```
DELAY  LDA A  #$05   ;在累加器A中加载05
LOOP   DEC A         ;累加器A减1
       BNE    LOOP   ;如果不等于零,则转移
       RTS           ;从子程序中返回
```

对于每次循环，包含了一系列的机器周期。当经过了 5 次循环时，延时程序所用周期为：

指　　令	周期数	总周期数	指　　令	周期数	总周期数
LDA A	2	2	BNE	4	20
DEC A	2	10	RTS	1	1

所以总延时为 33 个机器周期。如果一个机器周期为 1μs，则总延时为 33μs。为了更长的延时，初始化累加器 A 时可用更大的数。

一个 PIC 微处理器的延时子程序为：

291

```
        movlw  Value  ;加载所需的计数值
        movwf  Count  ;循环计数器
Delay   decfsz Count  ;递减计数器
        goto   Delay  ;循环
```

decfsz 指令用了 1 个周期，goto 指令用了两个周期。这个循环将重复（count-1）次。此外，movlw 和 movwf 指令各用了一个周期，当 count 等于 1 时 decfsz 又用了两个周期。所以总周期数为：

指令周期数 =3(计数值 -1) +4

如果每个指令周期需要 4 个时钟周期，则该子程序的延时周期数为：

时钟周期数 =4[3(计数值 -1) +4]

对于 4MHz 的时钟，每个时钟周期为 $1/(4\times10^6)$s。

通常只用上述单个循环得到的延时是不够的。一种实现更长时间的延时方法为嵌套循环。图 11-6 是嵌套循环延时的流程图。内部循环与上述的单循环相同。在循环结束且零标志位被设置前，寄存器 E 将会被递减 255 次。因此，外层循环会使内循环重复执行，直到寄存器 D 递减到 0。如果循环计数寄存器 D 的初始值设置为 140，则延时时间将为 140 × 2. 298 =321. 71ms。

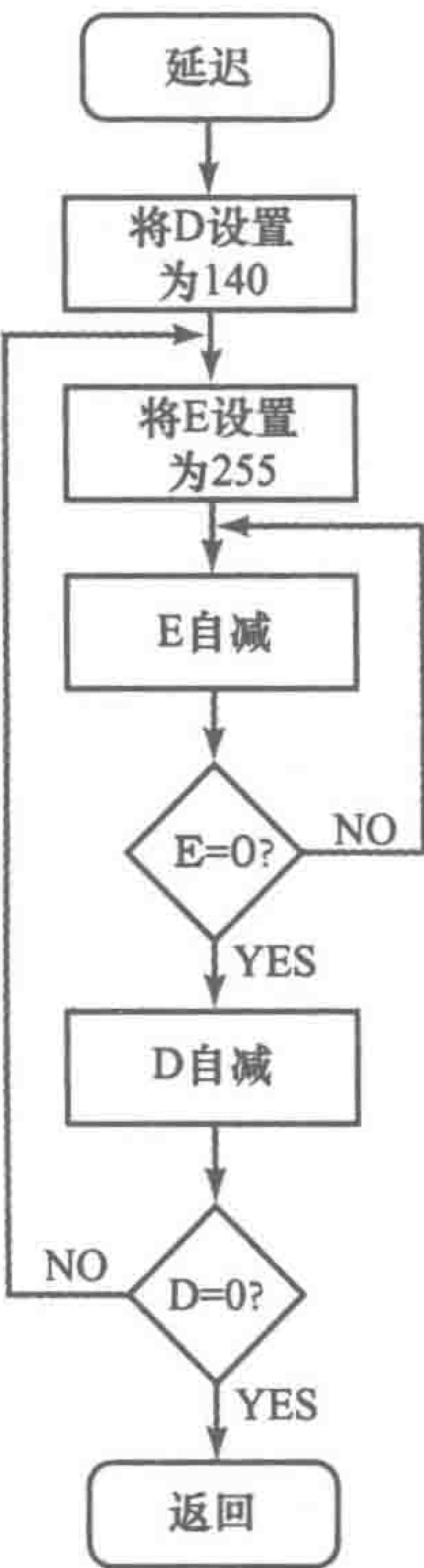


图 11-6　使用嵌套循环来实现延迟功能

程序为：

```
Delay  MOV  D,8CH   ;设置 D 为 8CH,即 140
OLOOP  MOV  E,FFH   ;设置 E 为 FFH,即 255
ILOOP  DEC  E       ;E 减 1,即内部计数值
       JNZ  ILOOP   ;重复内部循环 255 次
       DEC  D       ;D 减 1,即外部循环计数值
       JNZ  OLOOP   ;重复外部循环换 140 次
```

以下是一些包含延时子程序的例子：

问题：重复开关 LED 灯。

使用循环子程序 DELAY 来提供延时。微处理器用有限的时间来处理循环中的指令，并完成该循环。程序的结构为：

```
1)if LED on
     关 LED
     当 LED 关上时,执行子程序 TIME_DELAY
2)ELSE
     开 LED
     执行子程序 TIME_DELAY
```

```
子程序:TIME_DELAY
执行一条指令、多条指令、一个循环或双重循环取决于需要的延时长短。
```

因为延时长度的需要，所以可以使用双重循环来延时。对于 Intel 8051 编程，可以使用指令 DJNZ 减1，并在结果不为0时跳转。即它递减第一个操作数指定的值，如果结果不为0跳转到第二个操作数。LED 被连到微处理器端口 1 的第 0 位。于是，Intel 8051 的汇编指令程序为：

292

```
FLAG     EQU    0FH            ;在 LED 为 ON 时设置标志位
         ORG    8000H
START    JB     FLAG,LED_OFF   ;如果 FLAG=1 时,跳转,即 LED ON
         SETB   FLAG           ;否则置位 FLAG
         CLR    P1.0           ;LED ON
         LCALL  DELAY          ;调用延时子程序
         SJMP   START          ;跳至 START
LED_OFF  CLR    FLAG           ;清零 LED 开 FLAG,表明 LED 是关的
         SETB   P1.0           ;关 LED
         LCALL  DELAY          ;调出延时子程序
         LJMP   START          ;跳至 START
DELAY    MOV    R0,#0FFH       ;外部循环延时值
ILOOP    MOV    R1,#0FFH       ;内部循环延时值
OLOOP    DJNZ   R1,ILOOP       ;等待内部循环结束
         DJNZ   R0,OLOOP       ;等待外部循环结束
         RET                   ;从子程序返回
         END
```

问题：依次点亮 8 盏 LED。

循环移位指令可用于依次点亮 LED，于是我们可初始化一个字节模式 0000 0001，通过循环移位给出 0000 0011、0000 0111 等。以下是摩托罗拉 68HC11 汇编语言程序，LED 与端口 B 相连，程序中编入了一个短延时：

```
COUNT   EQU    8          ;计数值指定需要的循环数,即打开的位数
FIRST   EQU    %00000001  ;开启 0 位
PORTB   EQU    $1004      ;端口 B 的地址
        ORG    $C000
        LDAA   #FIRST     ;加载初始值
        LDAB   #COUNT     ;加载计数值
LOOP    STAA   PROTB      ;打开位 1 和 LED 1
        JSR    DELAY      ;跳至延时子程序
        SEC               ;进位位 C 设置为 1,并循环移入最低位,保持该位为 1
        ROLA              ;向左循环移位
        DECB              ;计数值减 1
        BNE    LOOP       ;跳转到 LOOP 8 次
DELAY   RTS               ;简单的短延时
        END
```

11.5 查找表

变址寻址可用于程序在表中查找数值。例如，在确定整数的平方数时，作为代替确定整数平方的计算方法，可在平方表中查找对应的特定整数的平方值。当某个关系是非线性或者不能通过简单的算术方程描述时，查找表法特别有用。例如，在 1.7.2 节中提到的引擎管理系统中，点火时间设置值是曲轴的角度和进气歧管压力的函数。这里，微处理器必须给出一个依赖于速度传感器和曲柄传感器输入信号的时间信号。

293

为了说明查找表的用法，考虑确定整数的平方值问题。我们可在程序存储器中放置整数0、1、2、3、4、5、6等的平方表，并在连续地址中有平方值0、1、4、9、16、25、36等项目。如果某个数的平方是4，则这个数就变为表中数据的变址寻址的变址，第一项是变址0。程序在表的基地址上加变址来找到对应整数的平方项的地址。于是我们有：

变址	0	1	2	3	4	5	6
表项	0	1	4	9	16	25	36

例如，对于摩托罗拉68HC11微控制器，也许我们有如下查找表程序来确定平方值：

```
REGBAS  EQU   $B600    ;表的基地址
        ORG   $E000
        LDAB  $20      ;在累加器B中加载要平方的数
        LDX   #REGBAS  ;指向表
        ABX            ;在变址寄存器X上加累加器B的内容
        LDAA  $00,X    ;在累加器A中加载变址寻址的值
```

我们还可用伪操作指令FDB加载表到存储器：

```
ORG  $B600
FDB  $00,$01,$04,$09  ;给出保留存储区块的值
```

对于Intel 8051微控制器，指令MOVC A，@A+DPTR根据DPTP与累加器A的和决定的存储器地址，提取数值并储存到累加器A中。该指令可用于在表中查找数据，数据指针DPTR初始化为表的开始。作为一个例子，假设我们想用一个表把温度的摄氏标度转换为华氏标度。程序涉及的温度参数传递需要转换子程序，所以可包含如下的指令：

```
          MOV    A,#NUM       ;加载将要转换的值
          CALL   LOOK_UP      ;调用LOOK_UP子程序
LOOK_UP   MOV    DPTR,#TEMP   ;指向表
          MOVC   A,@ A + DPTR ;从表中获取数据
          RET                 ;从子程序中返回
TEMP      DB     32,34,36,    ;表中的值
                 37,39,41,
                 43,45
```

294

另一个应用表的例子是序列数据输出。这可能是一个操作交通灯的序列，给出红、红+黄、绿、黄的点亮顺序。当RD0有输出时，红灯点亮；RD1有输出时，点亮黄灯；RD2有输出时，点亮绿灯。数据表为：

	红	红+黄	绿	黄
变址	0	1	2	3
	0000 0001	0000 0011	0000 0100	0000 0010

11.5.1　步进电动机的控制延时

为了操作控制步进电动机，在两个指令之间必须使用延时，使在下一条程序指令执行前电动机有时间前进一步。一个产生连续步进脉冲时序的程序可用如下的算法：

1）开始。
2）规定获得所需步进序列的输出时序。
3）设置到初始步进位置。
4）前进一步。
5）跳转到延时程序，使该步有时间完成。
6）判断当前一步是否是一个完整旋转的步进时序的最后一步？如果不是，继续下一步；否则，循环回步骤3）。
7）继续，直到结束。

以下是一个采用了整步模式的步进电动机的程序，用微处理器 M68HC11 控制，输出端口为 PB0、PB1、PB2、PB3。一个“查找”表用于输出代码序列，按照步进的时序输出来驱动步进电动机。

整步方式步进电动机操作所需的编码序列是 A、9、5、6，以 A 开始的这些值组成了一个序列，而指针必须要在这个表中查找它们。FCB 是“定义字节常量”的操作码，用于初始化表格的数据字节。

端口 B 的输出					
步骤	PB0	PB1	PB2	PB3	代码
1	1	0	1	0	A
2	1	0	0	1	9
3	0	1	0	1	5
4	0	1	1	0	6
1	1	0	1	0	A

```
BASE    EQU    $1000
PORTB   EQU    $4                   ;输出端口
TFLG    EQU    $23                  ;定时器中断标志寄存器1
TCNT    EQU    $0E                  ;定时器计数器寄存器
TOC2    EQU    $18                  ;输出比较 2 寄存器
TEN_MS  EQU    20000                ;10ms 时钟

        ORG    $0000
STTBL   FCB    $A                   ;这是一个查找表
        FCB    $9
        FCB    $5
        FCB    $6
ENDTBL  FCB    $A                   ;查找表结束

        ORG    $C000
        LDX    #BASE
        LDAA   #$80
        STAA   TFLG1,X              ;清空标志
START   LDY    #STTBL
BEG     LDAA   0,Y                  ;在表的第一个位置开始
        STAA   PORTB,X
        JSR    DELAY                ;跳转至延时
        INY                         ;表的变址寄存器 Y 递增
        CPY    #ENTBL               ;是否是表终点
        BNE    BEG                  ;如果不是,转移到 BEG
        BRA    START                ;如果是,再跳到 START
DELAY   LDD    TCNT,X
        ADDD   #TEN_MS              ;加上 10ms 延时
        STD    TOC2,X
HERE    BRCLR  TRLG1,X, $80,HERE    ;等待,直到延时结束
        LDAA   #$80
        STAA   TFLG1,X              ;清空标志
        RTS
```

注意，在上述程序中，标号 TEN_MS 中有一个空格下划线表明 TEN 和 MS 都是同一标号的部分。

在上述程序中延时是通过微处理器中的定时器模块来实现的。考虑10ms的延时。对于时钟频率为2MHz的微处理器系统，10ms的延时相当于20 000个时钟周期。因此，为了得到上述延时，需要将TCNT寄存器中的当前值与20 000个周期相加，再与TOC2寄存器中的值相加。

11.6　嵌入式系统

微处理器和微型控制器常常“嵌入”系统中，这样可以实现控制。例如，现代家用洗衣机中有嵌入式微型控制器，它已经用不同的洗衣程序来编程。所有的机器操作者只需通过开关选取需要的洗衣程序，该程序就被执行了。而操作者不必为微控制器编程。**嵌入式系统**（embedded system）用于基于微型处理器的系统，用来控制一种功能或一系列功能，而不是由系统用户进行编程。程序已由生产厂家编好，并“烧入”存储器系统，不能被系统用户改变。

295~296

11.6.1　嵌入式程序

在嵌入式系统中，生产厂家用ROM来存储程序。这种方法只有在有大量芯片需求的情况下才是经济的。另外，对于原型机或小批量应用，程序可以加载到应用硬件的EPROM/EEPROM中。下面说明如何对微控制器的EPROM/EEPROM进行编程。

例如，为了对Intel 8051微控制器的EPROM编程，需要图11-7a所示的线路。必须有4~6MHz的晶振输入。步骤概述如下：

1）需编程的EPROM的地址在0000H~0FFFH之间，使用端口P1和端口P2的引脚P2.0~P2.3，同时需编程的代码字节写入到地址端口P0。

2）引脚P2.7、RST、ALE应保持高电平，引脚P2.6和PSEN为低电平。引脚P2.4和P2.5是高电平还是低电平没关系。

3）引脚EA/V_{pp}在ALE输入编程脉冲前保持逻辑高电平，之后提高至+21V，ALE被输入50ms的低电平脉冲，把程序代码字节写入到地址单元，然后给EA返回逻辑高电平。

程序的验证，即读出程序的内容，是通过图11-7b中的线路实现的。

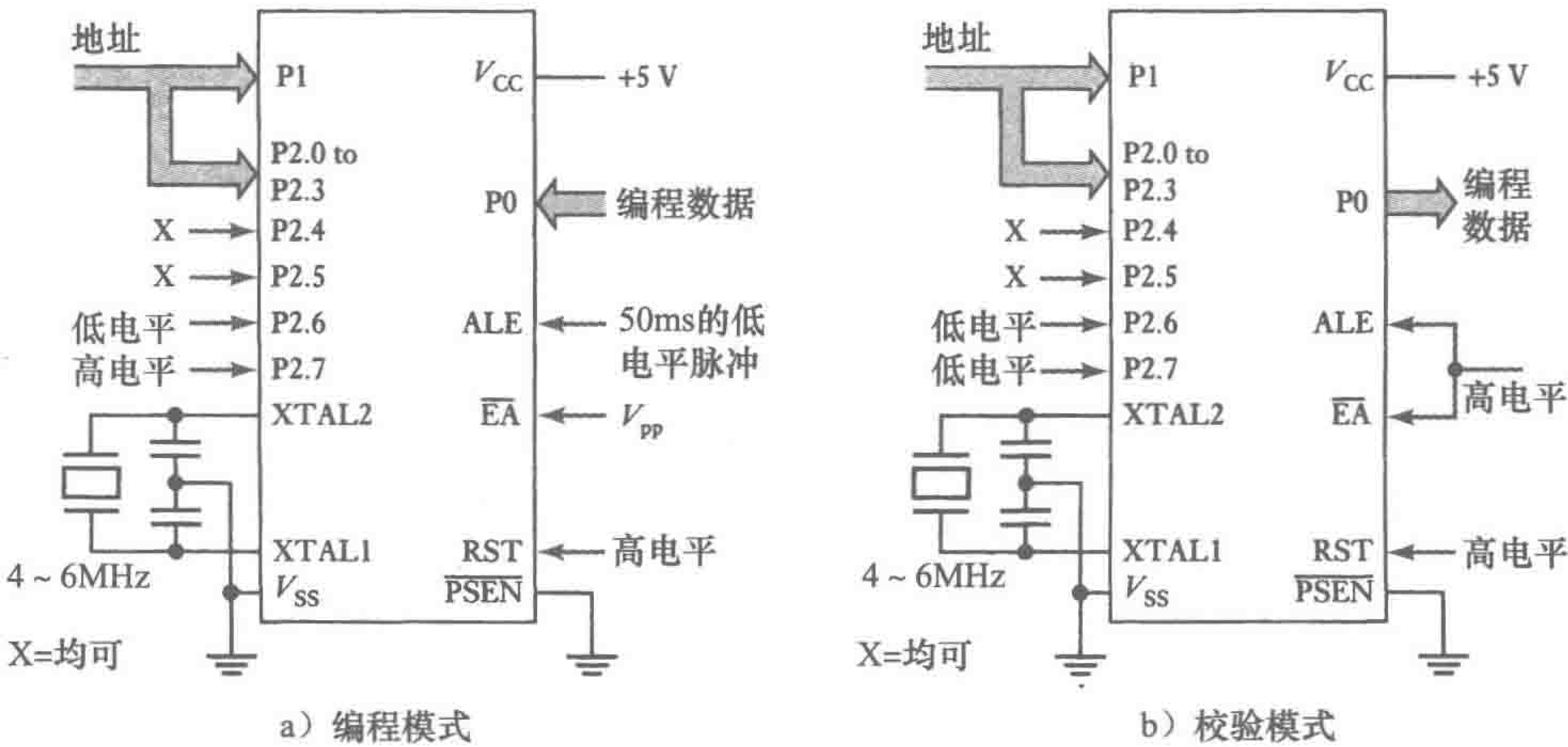


图11-7　Intel 8051微控制器

1）读出的程序地址为端口P1和端口P2的引脚P2.0~P2.3。

2）端口EA/V_{pp}、RST、ALE应保持为高电平，引脚P2.7、P2.6和PSEN为低电平。引脚

P2.4 和 P2.5 是高电平还是低电平没关系。

3）地址单元中的内容在端口 P0 输出。

可对一位安全位进行编程来禁止用任何外部方法对片内程序存储器的存取。安全位一旦写入，只有将程序存储器全部擦除时，安全位才能被擦除。该过程与正常编程写入的过程相似，但引脚 P2.6 要加高电平（见图 11-7a）。可以使用紫外线照射来擦除。因为太阳光和荧光灯中含有一些紫外线，应该避免长时间暴露在紫外线下（在太阳光下约为一星期，在室内级荧光灯下约为三年），而且芯片的窗口应该被不透明的标签遮盖。

297

摩托罗拉 68HC11 微控制器有一个内置电擦除可编程只读存储器（EEPROM）。EEPROM 的地址在 $B600 ~ $B7FF 之间。与 EPROM 一样，当所有的位为 1 时，就擦除一字节，编程时使一些特殊的位为 0。通过设置 CONFIG 寄存器中的 EEON 位为 1 来使能 EEPROM（见图 11-8），设置为 0 时禁止。程序的编写由 EEPROM 编程寄存器（PPROG）控制（见图 11-8）。

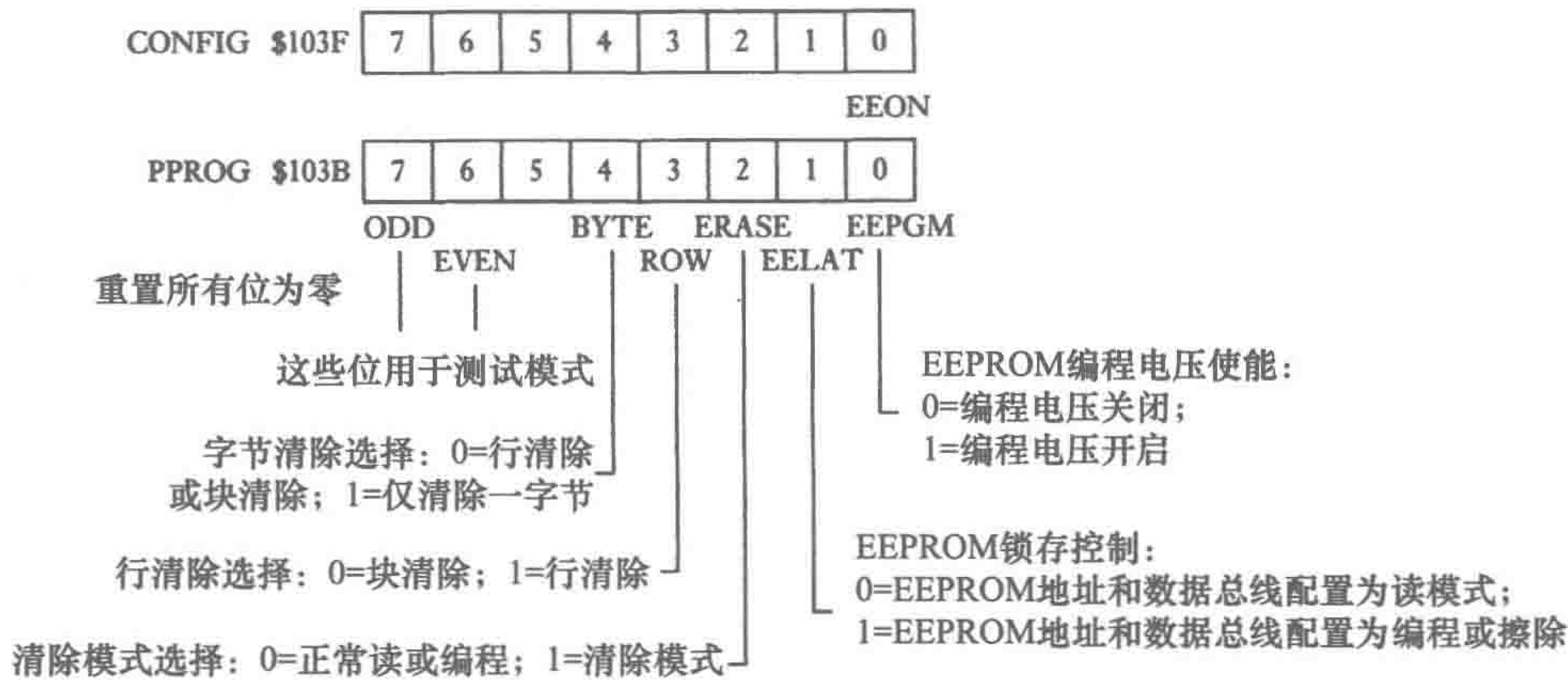


图 11-8　CONFIG 和 PPROG

编程过程概述如下：

1）写入 PPROG 寄存器设置 EELAT 位为 1 进行编程。
2）向选择的 EEPROM 地址写入数据。这一操作锁存地址和将要编写的数据。
3）写入 PPROG 寄存器设置 EEPGM 位为 1，打开编程电压。
4）延时 10ms。
5）写入 PRROG 寄存器为 0，关闭所有的位。

下面是用于 MC68HC11 汇编语言编程的子程序：

```
EELAT  EQU  % 00000010  ;EELAT 位
EEPGM  EQU  % 00000001  ;EEPGM 位
PPROG  EQU  $1028       ;PPROG 寄存器的地址

EEPROG
       PSHB
       LDAB #EELAT
       STAB PPROG       ;设置 EELAT = 1, EEPGM = 0
       STAA 0,X         ;把数据 X 存入 EEPROM 地址内
       LDAB #% 00000011
       STAB PPROG       ;设置 EELAT = 1, EEPGM = 1
       JSR  DELAY_10    ;跳转至 10ms 延时子程序
        CLR PPROG       ;清空所有 PPROG 位，并返回读取模式
       PULB
```

298

```
        RTS

;提供约 10ms 延时的子程序
DELAY_10
        PSHX
        LDX  #2500        ;计数 20 000 周期
DELAY   DEX
        BNE  DELAY
        PULX
        RTS
```

擦除的步骤：

1)写入 PPROG 寄存器，选择擦除 1 字节、1 行或全部 EEPROM。
2)在要擦除的范围内写入一个 EEPROM 地址。
3)向 PPROG 寄存器中写入 1 来打开 EEPGM 位，因此打开擦除电压。
4)延时 10ms。
5)向 PPROG 寄存器中写入 0，关闭所有的位。

针对内置 EEPROM 的 PIC 微控制器，向其写入数据的程序为（见图 11-9）：

```
bcf       STATUS,RP0              ;变到 BANK0
mov.f     Data,w                  ;加载要写入的数据
movwf     EEDATA
movf      Addr,w                  ;加载写数据地址
movwf     EEADR
bsf       STATUS,RP0              ;变到 BANK1 组
bcf       INTCON,GIE              ;禁止中断
dsf       EECON1,WREN             ;允许写
movlw     55h                     ;允许写的特殊顺序
movwf     EECON2
movlw     0AAh
movwf     EECON2
bsf       EECON1,WR               ;初始化写周期
bsf       INTCON,GIE              ;重新允许中断
btfsc EECON,WR
EE_EXIT                           ;检查写是否完成
goto      EE_EXIT                 ;如果未完成，重试
bsf       EECON,WREN              ;EEPROM 写完成
```

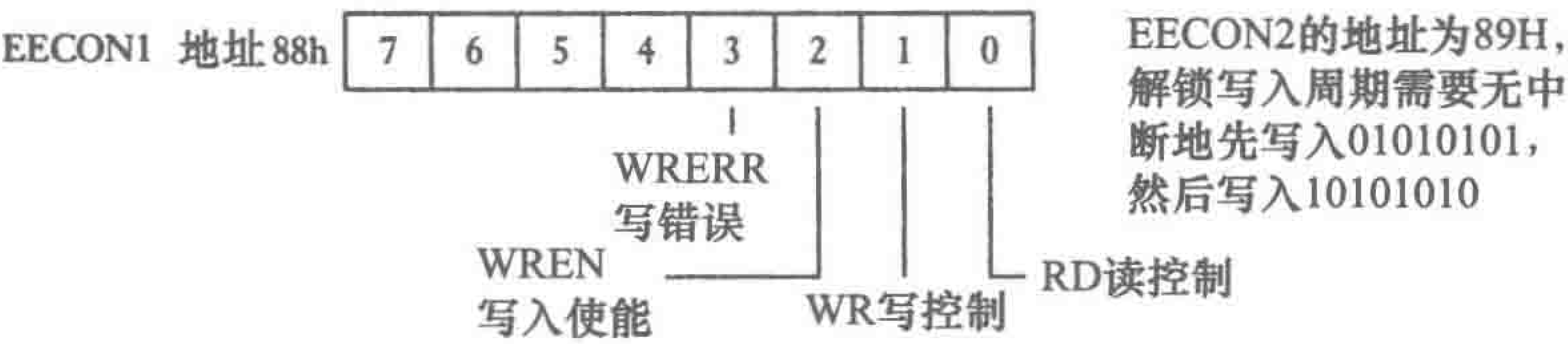


图 11-9　EECON 寄存器

299

本章小结

微处理器可以识别的一系列指令称为**指令集**。执行某一特殊任务所需要的一系列指令称为**程序**。

微处理器以二进制码工作。以二进制码写的指令是**机器码**。不是直接用二进制码，而是用简单、可读术语代替的速记代码称为**助记码**。助记码是一种辅助记忆的代码。这样的代码称为

汇编语言。汇编语言程序包含一系列语句，每行一句，每行语句包含 1 ~ 4 个字段：标号、操作码、操作数和注释。**标号**是指访问存储器的一个特定地址的名字。**操作码**是指怎样操作一个数据。**操作数**包含被操作的数据的地址。**注释**字段允许程序员包含注解，这可以使程序更容易被读者所理解。

习题

11.1 用以下从生产厂家的指令集（6800）中摘出的部分，确定带进位的加法所需要的机器码：(a) 立即寻址模式；(b) 直接寻址模式。

		寻址模式					
		立即（IMMED）			直接（DIRECT）		
操作	助记符	OP	~	#	OP	~	#
带进位的加法	ADC A	89	2	2	99	3	2

11.2 在摩托罗拉 6800 处理器的指令集中，清零操作在隐含寻址模式中只有一个入口。这一点的意义是什么？

11.3 对如下功能，摩托罗拉 6800 对应的助记符是什么：(a) 清除寄存器 A；(b) 存储累加器 A；(c) 加载累加器 A；(d) 比较累加器；(e) 加载变址寄存器？

11.4 写一行汇编语言程序：(a) 累加器 A 中加载 20(hex)；(b) 累加器 A 减 1；(c) 清空地址 $0020；(d) 把地址 $0020 中的数加到累加器 A 中。

11.5 解释由下列指令所指的操作：(a) STA B $35；(b) LDA A #$F2；(c) CLC；(d) INC A；(e) CMP A #$C5；(f) CLR $2000；(g) JMP 05，X。

11.6 写汇编语言程序：

(a) 用存储器地址 0060 中的十六进制数减去存储器地址为 0050 中的十六进制数，结果保存在 0070 中。

(b) 两个地址分别为 0020、0021 中的 8 位二进制数相乘，所得 8 位二进制数的积存储在地址 0022 中。

(c) 在 0020 开始的地址中存入十六进制的 0 ~ 10。

(d) 把从 $2000 地址开始的 32 个数移至 $3000 开始的地址中。

11.7 用汇编语言写一个可以设置为任意长度延时的子程序。

11.8 用汇编语言写一个程序，如果从开关传感器输入到地址 2000 中的值是高电平，程序跳转至从地址 3000 开始的程序，如果是低电平程序继续执行。

300 ~ 301

第12章 C 语 言

本章目的

通过本章的学习，读者将能够：

- 理解C语言程序的主要特点。
- 使用C语言编写简单的微控制器程序。

12.1 为什么选择C语言

本章的目的在于介绍C语言和C语言的编程。C语言是一种高级语言，常替代汇编语言（见第11章）用于编写微处理器的程序。与汇编语言相比，C语言更易于使用，且对不同的微处理器具有可移植性。但C语言需要相应的编译器才能够将程序翻译成相应微处理器的机器语言。汇编语言因微处理器型号不同而异，而C语言是标准化的，其标准由美国国家标准协会（American National Standards Institute，ANSI）制定。

12.2 程序结构

图12-1包含了一个C程序的主要元素。程序开头是预处理命令，它调用标准文件；其后是主函数。在主函数内，有其他的函数，这些函数作为子程序调用。每一个函数内都有多条语句。

302

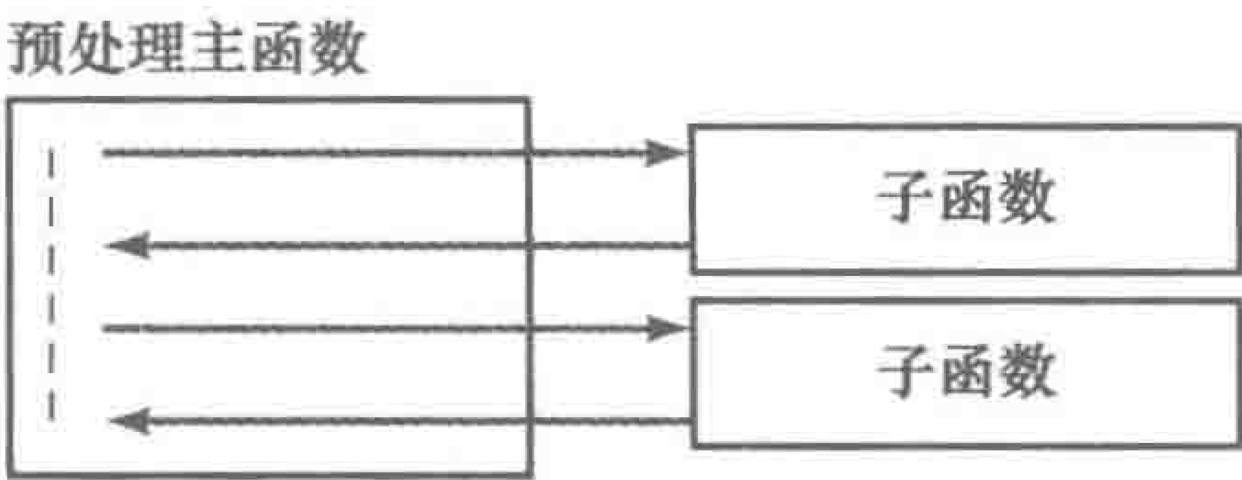


图12-1 C语言程序的结构

12.2.1 主要特征

下面描述C语言中的主要特征。注意：在C程序中，空格和回车仅仅使程序便于阅读，在编译过程中会忽略。

1. 关键字

在C语言中，有些单词被程序保留并赋予了特殊意义，不可改变。比如，int表示整型值，if表示程序可以根据if后表达式的真、假来选择不同的语句运行。C语言要求所有的关键字都用小写字母。下面列出了一些ANSI制定的C语言关键字：

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

2. 语句

语句组成程序，每一条语句以分号（;）结尾。语句可以通过大括号{}分成多组。以两条语句组成的组如下：

```
{
    语句1;
    语句2;
}
```

3. 函数

函数可定义为一组独立的程序代码，其包含一系列特定的操作，且具有一个函数名，用于被调用（类似于汇编语言中的子程序）。一个函数由一个函数名和一个小括号组成，如 name()。小括号中包含变量（函数的参数变量是当一个函数被调用时传递给函数的值）。调用函数时，只要直接在程序中写上函数名即可。如

```
printf("Mechatronics");
```

表示将单词 Mechatronics 传递给函数 printf()。printf() 是一个被事先写好的，被预处理命令调用了的函数，因此，单词可以显示在屏幕上。为了表示参数中的字母是一个整体，比如上例中的单词 Mechatronics，在字母两侧用双引号括起来。

303

4. 返回

函数可能需要将一个值返回给调用程序中。被调用函数的**类型**应在函数名字的前面声明。当函数执行完后，将指定类型的值返回给调用函数。比如，int main() 表示从主函数返回一个整数。当然，如果函数没有返回值，可以定义函数的返回值是 void，如 void main（void）。一般而言，头文件会包含返回信息，所以头文件定义的函数在程序中不用再定义返回值类型。

如果想将一个值从函数体中返回到调用点，使用关键字 return，如：

```
return result;
```

return 语句结束了该函数。

5. 标准库函数

C 语言包提供了许多函数库，其中预先定义了许多已经写好的常用函数，使用时直接调用即可。如果想要使用函数库中的函数，该库需要在头文件中声明。这样的库文件的例子如下：

math. h 包含数学函数

stdio. h 包含输入/输出函数

time. h 包含日期时间函数

例如，printf() 是 stdio. h 中的函数，它功能就是将结果输出到屏幕上。另一个函数 scanf() 是从键盘中读入数据。

6. 预处理程序

预处理程序是由**预处理命令**识别，并在编译前运行的程序，以#开头。如：

```
# include < >
```

读入的文件名写在尖括号（< >）内。当程序运行到此处时，指定的文件将导入程序中。常用于添加标准头文件的内容，使得许多声明和定义可以使用标准库函数。使用的程序段可能为：

```
# include <stdio.h>
```

为了说明上述内容，考虑下面的程序：

```
# include <stdio.h>
main( )
{
  printf("Mechatronics");
}
```

304

在运行主函数之前，文件 stdio. h 添加到程序中。因此，当主程序运行时，就可以使用函数 printf()，将单词 Mechatronics 输出到屏幕上。

另一种预处理命令为：

```
# define pi 3.14
```

该命令用程序中特殊的符号来定义一些值。因此，当遇到 pi 时，值 3. 14 就被调用。

又如：
```
# define square(x) (x)*(x)
```
当程序中出现 square(x) 时（这里的 x 可以是数或者表达式），会自动被（x）*（x）替换。

7. 主函数

每一个 C 程序都必须有一个 main() 函数。这个函数在程序运行时控制整个程序，并且是第一个被调用的函数。可以理解为整个程序的第一行语句，其他函数都可以被主函数调用，被调用的函数依次执行之后又返回到主函数中。主函数的一种形式如下：
```
void main(void)
```
这表明主函数在运行后将没有结果返回到主程序中，并且没有自变量。一般约定从主函数中返回 0 作为主函数结束的标志，如：
```
return 0;
```

8. 注释

/*和*/用于括起注释内容。因此我们可以这样编程：

/* 主程序如下 */

注释在编译时被忽略，仅用于方便程序理解。注释可以跨越多行，如：

/* 一个程序被用于

介绍如何编程 */

9. 变量

变量是一种被命名的存储单元，其存储的值是可变的。可以存储一个字符的变量用关键字 char 来定义。这样的变量有 8 位，常用于存储单个字符。有符号整数，即没有小数部分且有符号指示正、负的数，用 int 来定义。关键字 float 用于定义单精度浮点数，这些数是有小数部分的。关键字 double 用于定义双精度浮点数，其有效位数是 float 的两倍。声明变量类型时只要在变量名前加上变量类型就可以了，如：
```
int counter;
```
这就为变量 counter 定义了 int 类型。另一个例子为：
```
float x, y;
```
这定义了 x、y 两个变量都是浮点数的类型。

10. 赋值

赋值语句可以将等号右侧表达式的值赋值给等号左侧的变量。如：a=2 将 2 赋值给变量 a。

11. 算术运算符

常用算术运算符有加号 +、减号 -、乘号 *、除号/、求余%、自加 ++ 和自减 --。自加号将变量的值加 1，自减号将变量的值减 1。算术运算符的优先级与数学中的规定一致。如，2*4+6/2 的结果是 11。含有算术运算符的一个程序例子如下：

```
/* 计算圆的面积*/
#include <stdio.h>  /* 导入函数库*/
int radius, area;  /* 变量 radius 和 area 都是整数 int*/
int main(void)  /* 启动主函数,int 表示一个 int 值将会返回,
      void 表示主函数 main()没有参数*/
{
      printf("输入半径:");  /* "输入半径:"会被输出到屏幕上*/
      scanf("% d",&radius);  /* 从键盘上读入一个整数,并将之赋值给 radius*/
      area=3.14* radius* radius;  /* 计算面积*/
      printf("\n Area=% d",area);/* 在新的一行打印"Area=",并且输出结果*/
      return 0;  /* 返回到调用点*/
}
```

305

12. 关系运算符

关系运算符用于比较表达式，例如“x 与 y 是否相等”、“x 是否大于 10” 等。关系运算符有等于 == 、不等于! = 、小于 < 、小于等于 <= 、大于 > 、大于等于 >= 。注意， == 用于判断两个变量是否相等，而 = 用于赋值。比如，我们有一个问题“x 是否等于 2”，那么它相应的表达就是（a == 2）。

306

13. 逻辑运算符

逻辑运算符如下：

运　算	符　号
与	&&
或	||
取反	!

注意，在 C 语言中，true 等于 1，false 等于 0。

14. 位运算

位运算符在处理操作数时，是将其作为一系列独立的位（bit），而不是一个数值。并且位运算只能适用于整数，按每个操作数中对应的位进行比较。位运算符包括：

位运算	符　号
与	&
或	|
异或	^
取反	~
右移	>>
左移	<<

例如：

```
portA = portA | 0x0c;
```

前缀 0x 表示 0c 是十六进制数，用二进制表示为 0000 1100。与 portA 进行或运算后的值只是将 portA 的 2 和 3 位的值强制置 1，其他的位保持不变。

```
portA = portA ^ 1;
```

这个表达式使得除了第 0 位外所有的位都保持不变。如果 portA 第 0 位为 1，异或使其变为 0；如果为 0，就变成 1。

15. 字符串

一系列用双引号（""）括起的字符称为字符串。如该术语所表示的，双引号中的字符作为一个整体看待。如：

```
printf("Sum = %d", x);
```

小括号中的语句是传递给 print 函数的。这里有两个参数，以逗号分隔。前一个参数是双引号中的字符串，表示要输出的内容，%d 表示要显示的变量是一个十进制整数。其他格式的字符有：

307

%c　字符

%d　有符号十进制整数

%e　科学记数法

%f　十进制浮点数

%o　无符号八进制数

%s　字符串

%u　无符号十进制整数

%x　无符号十六进制数

%%　输出百分号%

另一自变量 x 表示要显示的值。

另一个例子为：

```
scanf("%d", &x);
```

这个语句从键盘读取一个十进制整数，并将其赋值给整型变量 x。变量 x 前的符号 & 是地址运算符。当 & 置于一个变量前时，地址运算符返回的是变量的地址。这条命令可以扫描数据并存储到给定的地址。

16. 转义字符

所谓转义字符，就不再是字面上的意思了，通常用于控制光标的移动，比如退格、Tab、回车等，当然还可以表示其他的操作。例如：

```
printf("\nSum = %d", d);
```

其中，\n 表示在新的一行输出结果。常用的转义字符有：

\a　计算机发出嘀的一声

\b　退格

\n　换行

\t　水平 TAB

\\　反斜线

\?　问号

\'　单引号

12.2.2　一个 C 程序的示例

下面用一个简单的程序来说明上述用法。

```
/* 一个简单的 C 程序*/
#include <stdio.h>
void main(void)
{
    Int a,b,c,d;/* a,b,c,d 是整数*/
    a=4;  /* a 赋值为 4*/
    b=3;  /* b 赋值为 3*/
    c=5;  /* c 赋值为 5*/
    d=a* b* c;  /* d 赋值为 a* b* c*/
    printf("a* b* c=% d\n",d);
}
```

308

语句 int a，b，c，d；声明了 a，b，c，d 为整数类型。语句 a=4，b=3，c=5 将初始值赋值给变量，=表示赋值。语句 d=a*b*c 表示 a，b，c 三者的乘积赋给 d。语句 printf("a*b*c=%d\ n",d) 中的 printf 是屏幕输出函数。其中包括%d，表示结果要转换为十进制数。所以输出为 a*b*c=60。字符 \n 在字符串尾表示输出后换行。

12.3　分支和循环

能够表示分支和循环的语句有 if、if/else、for、while 和 switch。

1. if

if 语句允许分支转移（如图 12-2a 所示）。例如，如果表达式是真，语句被执行；否则就不执行，程序运行下面的语句。

```
if (condition 1 = = condition  2);
printf ("\nCondition is OK.");
```

一个包含 if 语句的程序例子如下：

```
#include  <stdio.h>

int x, y;
main( )

{
    printf("\nInput an integer value for x: ");
    scanf("%d", &x);
    printf("\nInput an integer value for y: ");
    scanf(%d", &y);
    if( x = = y)
       printf("x is equal to y");
    if(x > y)
       printf("x is greater than y");
    if(x < y)
       printf("x is less than y");
       return 0;
}
```

屏幕上会显示“Input an integer value for x：”，然后需要输入一个值。之后，屏幕上会显示“Input an integer value for y：”，需要输入另一个。if 语句判断输入的值等于、大于还是小于另一个值，并将结果输出到屏幕。

2. if/else

if 语句能够与 else 语句组合。这使得一组语句在结果为真时可以执行，而另一组语句在结果为假时可以执行（如图 12-2b 所示）。例如：

```
#include <stdio.h>

main( )
{
  int temp;
  if(temp > 50)
     printf("Warning");
  else
    printf("System OK");
 }
```

3. for

循环 loop 用于描述这样一种结构：一系列语句一直在执行直到一个特定的条件达到某种状态（或者为真，或者为假）。如图 12-3a 所示。用 for 函数实现的循环结构，其一般的形式是：

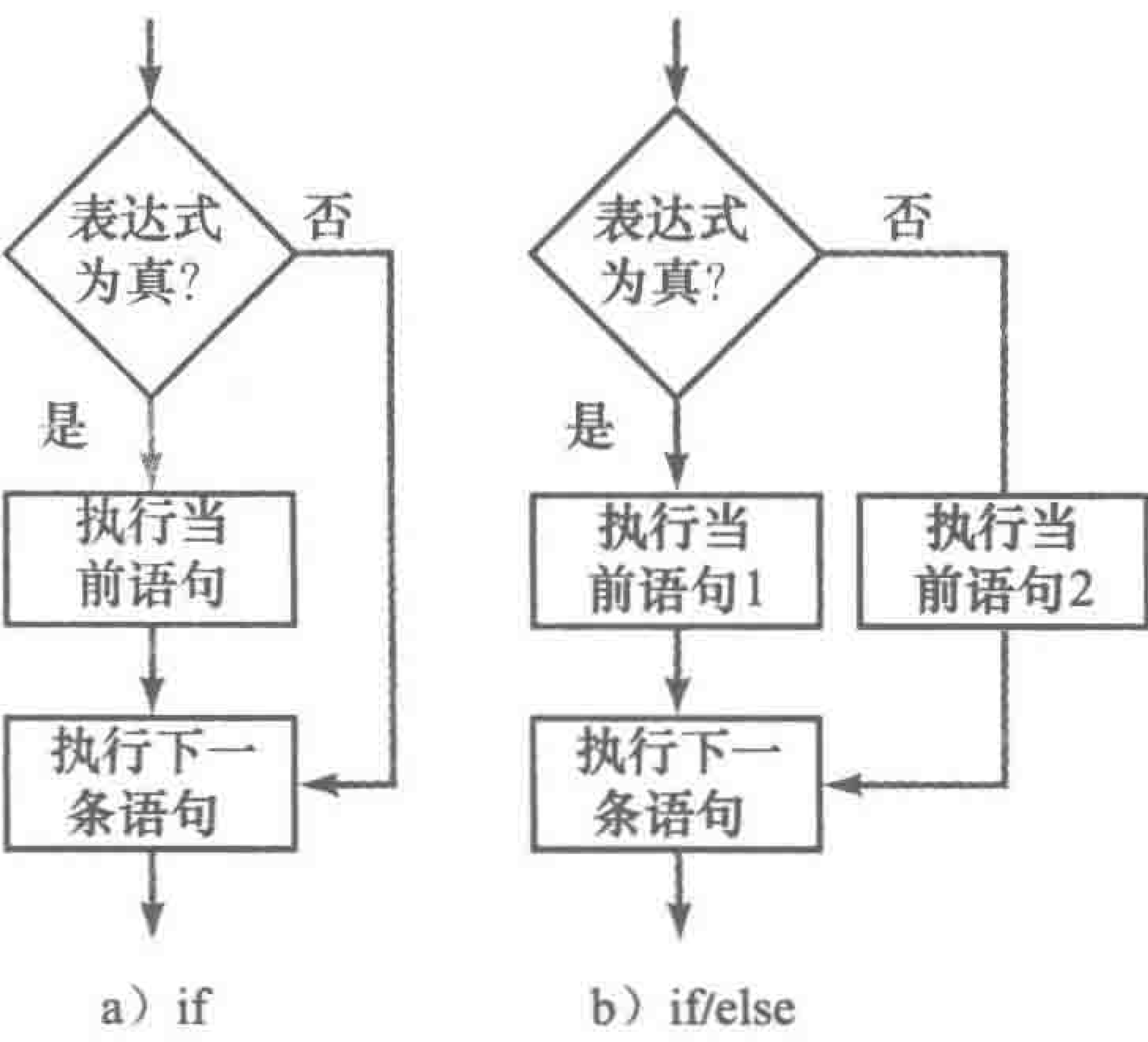


图 12-2　判断

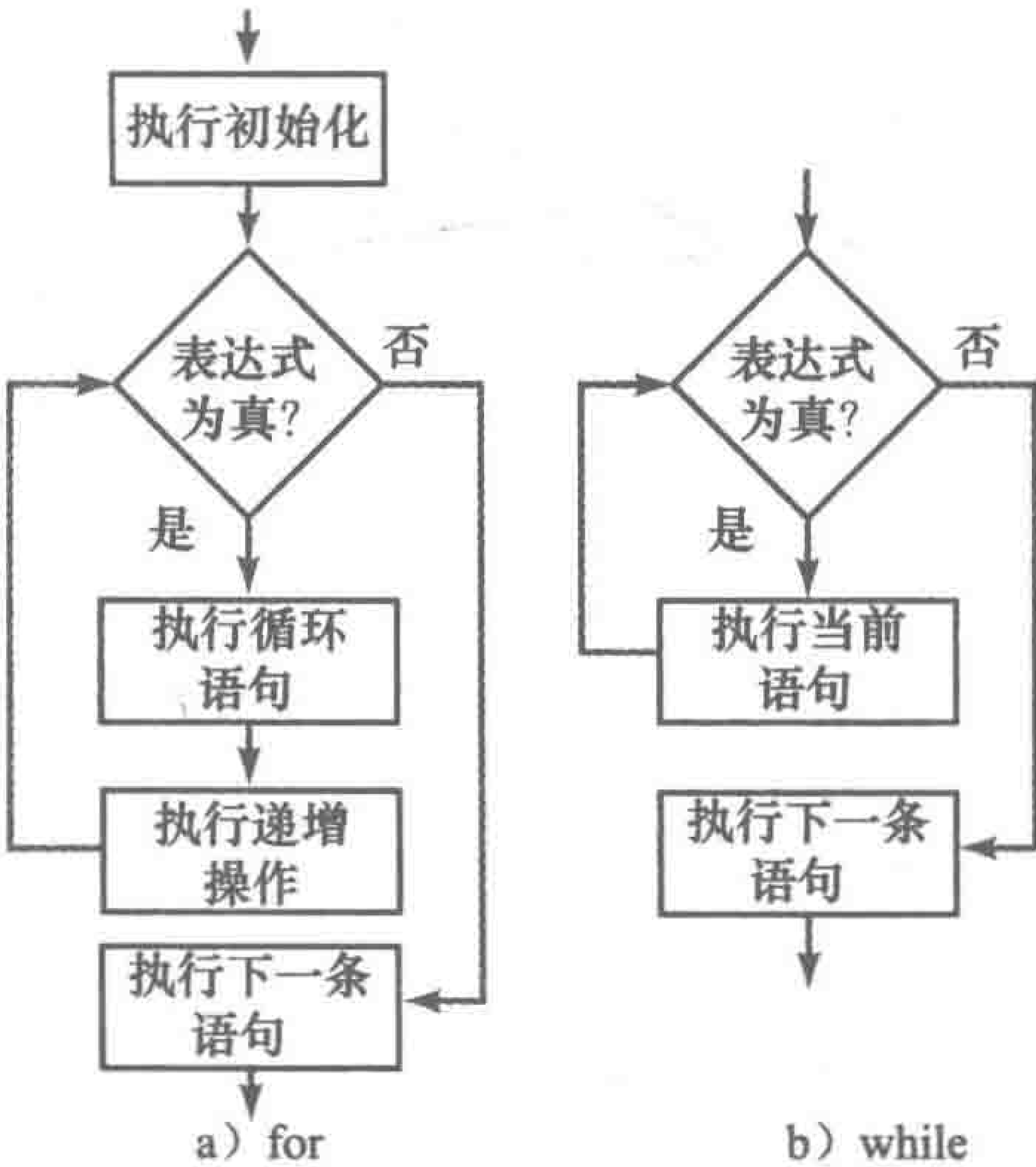


图 12-3　循环

```
for(初始表达式;测试表达式;循环变量增值)
循环体;
```

下面是一个例子：

```
#include <stdio.h>

int count;

main( )
{
  for(count = 0; count < 7; count ++)
  printf("\n%d", count);
}
```

初始时，count 的值为 0。此后不断加 1，只要 count < 7，就执行循环体内的语句。屏幕上显示的结果是 0 1 2 3 4 5 6，且每个数显示在分开的行上。

4. while

只要表达式是真的，while 语句允许循环一直执行（见图 12-3b）。当表达式为假时，程序跳出循环，执行之后的语句。下面的程序就是用 while 循环实现当 count < 7 时输出结果：

```
#include <stdio.h>

int count;
int main( );
{
   count = 1;
   while(count < 7)
      {
         printf("\n%d", count);
         count ++;
      }
      return 0;
}
```

309 ~ 311

输出结果是 0 1 2 3 4 5 6，且每个数显示在分开的行上。

5. switch

switch 语句允许程序选择不同的语句执行。其条件在括号内。可以执行的语句在 case 后，这需要确定条件的值是否等于 case 后的值。比如，当 case 1 发生时执行语句 1，当 case 2 发生时执行语句 2，等等。如果条件不等于任何 case 语句，default 就会被执行。一般而言，在每一个 case 语句中都会有一个 break，这样在执行完这个 case 的语句后可以跳出 switch，防止继续判断其他的 case。结构如下（如图 12-4 所示）：

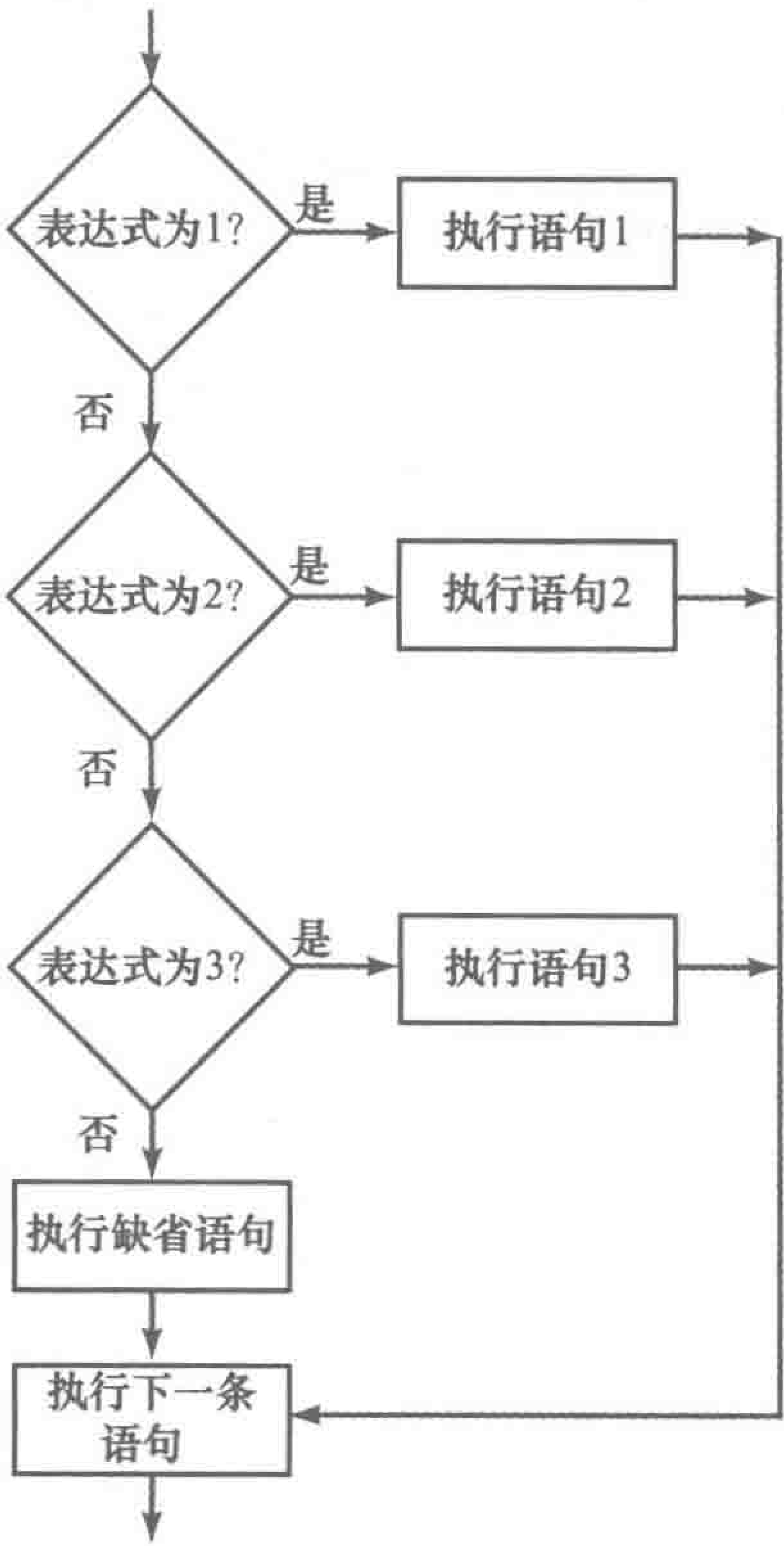


图 12-4　switch

```
switch(expression)
{
  case 1:
    statement 1;
  break;
  case 2:
    statement 2;
    break;
  case 3:
    statement 3;
    break;
  default:
    default statement;
}
next statement
```

下面是一个程序，判断键盘输入的数是 1、2 还是 3 并显示到屏幕上：

```
#include <stdio.h>
int main()
{
    int x;
    printf("输入一个数 0,1,2 或者 3:");
    scanf("% d",&x);
    switch(x)
    {
        case 1:
            printf("One");
            break;
        case 2:
            printf("Two");
            break;
        case 3:
            printf("Three");
            break;
        default:
            pirntf("Not 1,2 或 3");
    }
    return 0;
}
```

12.4 数组

假定我们需要记录一周内每天中午的气温，然后检索某天的气温。这就需要使用数组。数组是这样的一组存储位置：它们存储同样的数据类型，拥有同样的名字。为了定义一个 Temperature 数组来存储气温（float 类型），我们有如下的语句：

```
Float Temperature[7];
```

数组的长度直接在数组名后的方括号 [] 中声明。上例中，7 用于记录一周 7 天的数据。数组中的某个值用索引值来访问。第一个单元的下标是 0，第二个是 1，以此类推，一个长度为 n 的数组的最后一个单元的下标是 $n-1$。图 12-5a 展示的就是一个数组的组成形式。向数组中存储值可以像下面这样写：

312 ~ 313

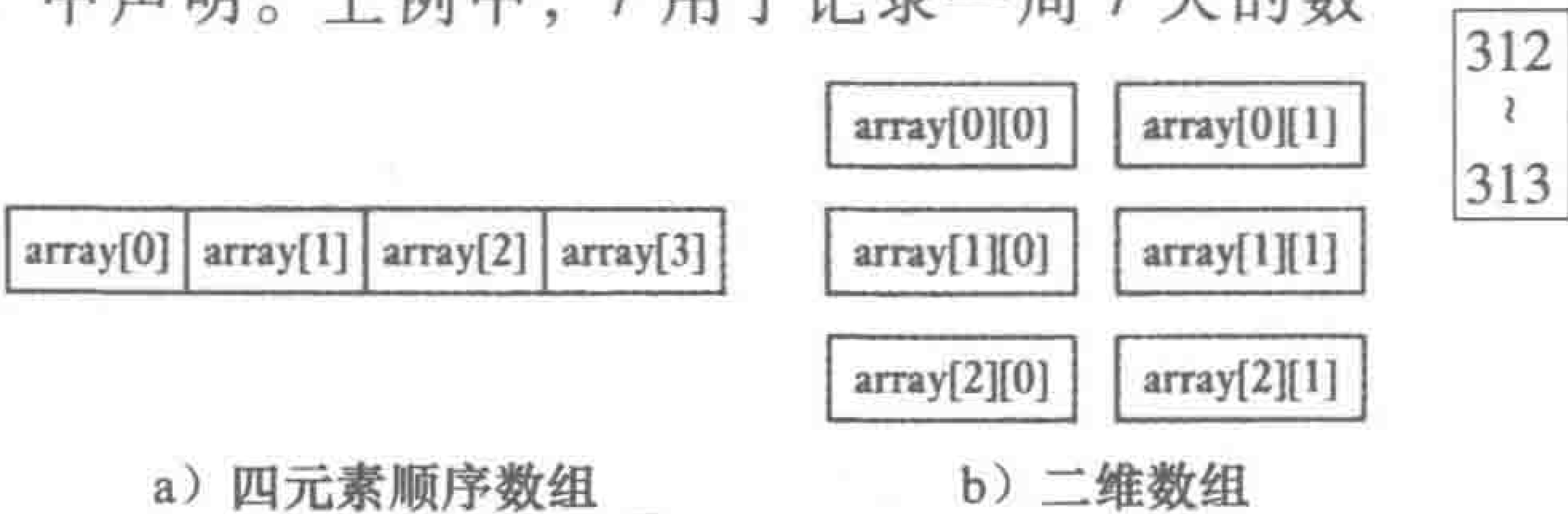


图 12-5 数组

```
Temperature[0] =22.1;
Temperature[1] =20.4;
等等。
```

如果需要使用 scanf() 给数组输入数据，那么在变量名前加 &，如：

```
scanf("% d", &Temperature[3]);
```

下面是一个用数组存储 0，1，2，3，4 并显示的程序：

```
#include <stdio.h>

int main(void)
{
  int sqrs[5];
```

```
  int x;

  for(x = 1; x<5; x++)
    sqrs[x – 1] = x * x;
  for(x = 0; x < 4; x++)
    printf("%d", sqrs[x]);

  return 0;
}
```

在定义数组时，我们可以直接给数组赋予初始值：

```
int arrya[7] = {10,12,15,11,10,14,12};
```

如果省略数组长度，编译器会自动创建一个恰好包括所有初始值的数组：

```
int array[ ] = {10,12,15,11,10,14,12};
```

也可以使用多维数组。例如，一组数据需要两维数组（见图12-5b），其中x表示行，y表示列，表示方法如下：

314

```
array[x][y];
```

12.5 指针

每一个存储单元都有唯一的地址，其作用是使存储的数据可以通过指针访问。**指针**是一种特殊的变量，它可以存储其他变量的地址。因此，如果变量p存储着变量x的地址，那么p就称为指向x的指针。如果x在存储器中的地址是100，那么p的值就是100。指针是变量，因此在使用前也要定义。其定义方式如下：

```
type *name;
```

*表示变量类型是指针。通常，指针名都有前缀p，即pname的形式。如：

```
int * pnumber;
```

初始化指针并给其一个指向的地址，我们可以用&，它是一个取地址操作符。用法如下：

```
Pointer = &variable;
```

下面一个小程序介绍了上面的用法：

```
#include <stdio.h>
int main(void)
{
     int * p,x;
     x =12;
     p = &x;  /* 将 x 的地址赋值给 p*/
     printf("% d",* p);/* 用指针输出 x 的值*/
     return 0;
}
```

程序在屏幕上显示12。上面用指针访问一个变量的值的方式称为间接访问。访问一个被指针指向的变量的过程称为指针的**解引用**，即取指针指向地址的内容。

12.5.1 指针的算术运算

指针变量能够进行算术运算，如：+、-、++和--。自加或者自减一个指针，其结果是指向数组中的下一个元素或者前一个元素。因此，自加指针以得到一个数组中的下一项，我们有如下的用法：

```
pa ++;/* 用自加符*/
```

或者：

```
pa = pa + 1;/* 加 1*/
```

315

12.5.2 指针和数组

指针可以用于访问数组中的元素，下面就是一个例子：

```
#include < stdio.h >
int main(void)
{
    int x[5] = {0,2,4,6,8};
    int * p;
    p = x;/* 将 x 的初始地址赋值给 p*/
    printf("% d % d",x[0],x[2]);
    return 0;
}
```

语句 printf("%d%d",x[0],x[2])；指向 x 提供的地址，因此在地址［0］和［2］的值被显示出来，结果就是 0 和 4，并在不同的行。

12.6 程序开发

开发程序的目的是生成一组可用于控制微处理器或微控制器的机器语言指令。这些指令称为**可执行文件**。为了得到这样的文件，我们需要进行下面的一系列过程：

1. 创建源代码

这就是用 C 语言编写组成程序的一系列语句。许多编译器都有编辑器，因此程序员只要从键盘键入源代码就可以了。其他的程序也可以达到相同的目的，如 Microsoft Windows 的记事本。使用文字处理器能够在格式错误的时候提示你，并且阻止进行程序编译，只有当保存的文件没有格式错误时，才能够编译。

2. 编译源代码

只要源代码写好了，程序员就可以使用编译器将其翻译成机器语言。在编译过程开始前，所有的预处理命令都会被执行。编译器能够在翻译过程中检测到一些不同形式的错误，并生成出错信息。有时一个简单的错误会导致一系列之后的错误。如果有错误通常需要回到编辑状态，重新编辑源代码。编译成功后，编译器会将生成的机器语言存放在另一个文件中。

3. 连接生成可执行文件

然后，编译器会将所有的生成代码连接起来，再与库函数连成整体，从而产生一个可执行文件。之后，程序保存为可执行文件。

316

12.6.1 头文件

预处理命令用在程序的开头，定义在程序中使用的函数。这样，这些函数就能通过简单的标号来调用。但是，为了避免编写一系列长长的代码来为每一个程序定义标准函数，我们可以用预处理命令来指定一个需要导入的文件，这个文件中包含了相关的标准函数。这时我们就只需要指出哪一个标准函数文件需要被编译器使用。因为这样的文件是写在一个程序的开头，所以被称为**头文件**。例如，<stdio.h>包含了标准输入和输出函数，如 gets（输入，即从一个设备中读入一行）、puts（输出，即向一个设备写入一行）和 scanf（读数据）；<math.h>包含数学函数，如 cos、sin、tan、exp（指数函数）和 sqrt（平方根函数）。

头文件同样适用于定义微控制器的寄存器和端口，使程序员在编写预处理命令时不用每次都定义寄存器和端口。因此，对于 Intel 8051 芯片，有头文件 <reg.51.h>，这定义了寄存器，例如端口 P0、P1、P2 和 P3，以及位寻址寄存器中的独立位，如 TCON 寄存器中的 TF1、TR1、TF0、TR0、IE1、IT1、IE0 和 IT0。因此我们可以只需要使用 P0，就能够实现端口 0 的输入/输

出。或者通过 TF1 来访问在 TCON 寄存器中的 TF1 位。类似地，对于摩托罗拉 M68HC11E9 芯片，头文件 <hc11e9.h>定义了寄存器，如 PORTA、PORTB、PORTC 和 PORTD，以及可位寻址寄存器的独立位，如寄存器 PIOC 中的 STAF、STAI、CWOM、HNDS、OIN、PLS、EGA 和 INVB。因此，我们只需用标号 PORTA 来编写指令实现端口 A 的输入/输出。一些库文件也可以提供帮助使用诸如小键盘、液晶显示等外部设备的程序。

为某个特定的微控制器编写的程序，通过改变头文件，可以轻松地移植到其他微控制器。库文件使 C 语言程序有了很高的可移植性。

12.7 程序的示例

下面的几个例子是用 C 语言为微控制器系统编写的程序。

12.7.1 电动机的起动与停止

利用 M68HC11 微控制器编写程序使直流电动机起动和停止。端口 C 用于输入，端口 B 通过合适的驱动器向电动机输出（见图 12-6）。起动按钮连接到 PC0，从 0 转变到 1 从而使电动机起动。停止按钮连接到 PC1，当从 1 转变到 0 时使电动机停止。端口 C 的数据方向寄存器 DDRC 应设置为 0，这样端口 C 才能设置为输入。

317

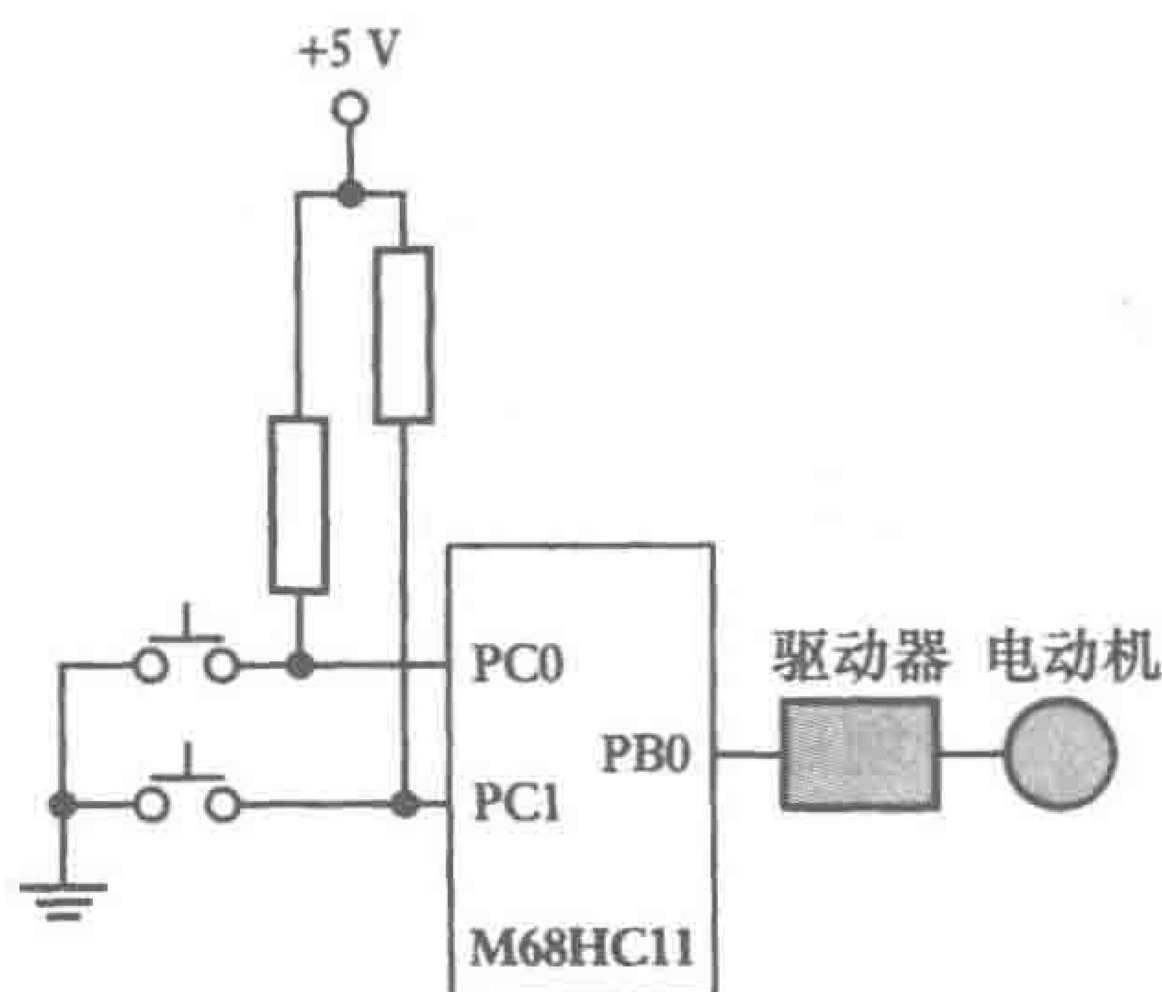


图 12-6 电动机控制

程序如下：

```
#include <hc11e9.h>/* 包含头文件*/
void main(void)
{
    PORTB.PB0 &=0;/* 初始化应保证电动机关闭*/
    DDRC=0;  /* 使端口 C 用于输入*/
    while(1)/* 当满足该状态时,不断重复*/
    {
        if (PORTC.PC0 ==0)/* 启动按钮是否被按下? */
            PORTB.PB0 |=1;/* 如果按下,启动命令输出*/
        else if (PORTC.PC1 ==0)/* 停止按钮是否被按下? */
            PORTB.PB0&=0;/* 如果按下,停止命令输出*/
    }
}
```

注意，| 是或运算符，只有在两个相应的位都为 0 时，结果才设置为 0，否则设置为 1。这用于开启或设置某一个值中的一个或多个位。因此 PORTB. PB0| =1 使得 1 与 PB0 中的值做或运算，从而起动电动机。这是一种同时改变一系列位值的有用方法。在 PORTB. PB0& =0 中，& 用于把 PB0 和 0 做与（AND）运算，将 0 赋值给 PORTB. PB0。

12.7.2 读 ADC 通道

考虑这样一个任务：编程控制微控制器 M68HC11，使得模拟数字转换器（ADC）的一个单一可以被通道。M68HC11 包括一个 8 通道多路开关、8 位逐次逼近的 ADC，其输入口为端口 E（见图 12-7）。ADC 的控制 - 状态寄存器 ADCTL 在第 7 位有转换完成标志 CCF，在其他位上控制多路转换开关和通道扫描。当 CCF =0 时，转换未完成；当 CCF =1 时，转换完成。模拟信号转变为数字信号（模数转换）是通过向 OPTION 寄存器里的模拟数字转换上电（ADPU）位写入 1 开始的。然而，ADC 在读取数值前至少要开启转换 100μs 以上。

为了把模拟输入转换到 PE0，ADCTL 寄存器中的前 4 位，如 CA、CB、CC 和 CD，必须全部设置为 0。如果只转换一个通道，第 5 位 SACN 和第 4 位 MULT 需要设定为 0。在开启 ADC 后，一个简单的读取某一个通道的程序要将 ADCTL 寄存器中的所有位都设置为 0，然后输入通道号，接着当 CCF =0 时读入输入信号。

因此，程序如下：

```
#include <hc11e9.h>/* 头文件*/
void main(void)
{
    unsigned int k;/* 用于输入通道号*/
    OPTION=0;  /* 这条和下条开启 ADC*/
    OPTION.ADPU=1;
    ADCTL=0;  /* 所有位清零*/
    ADCTL|=k;  /* 输入通道号,准备读入数据*/
    while(ADTCL.CCF==0);
    return ADR1;/* 返回转换的值到地址 1*/
}
```

318~319

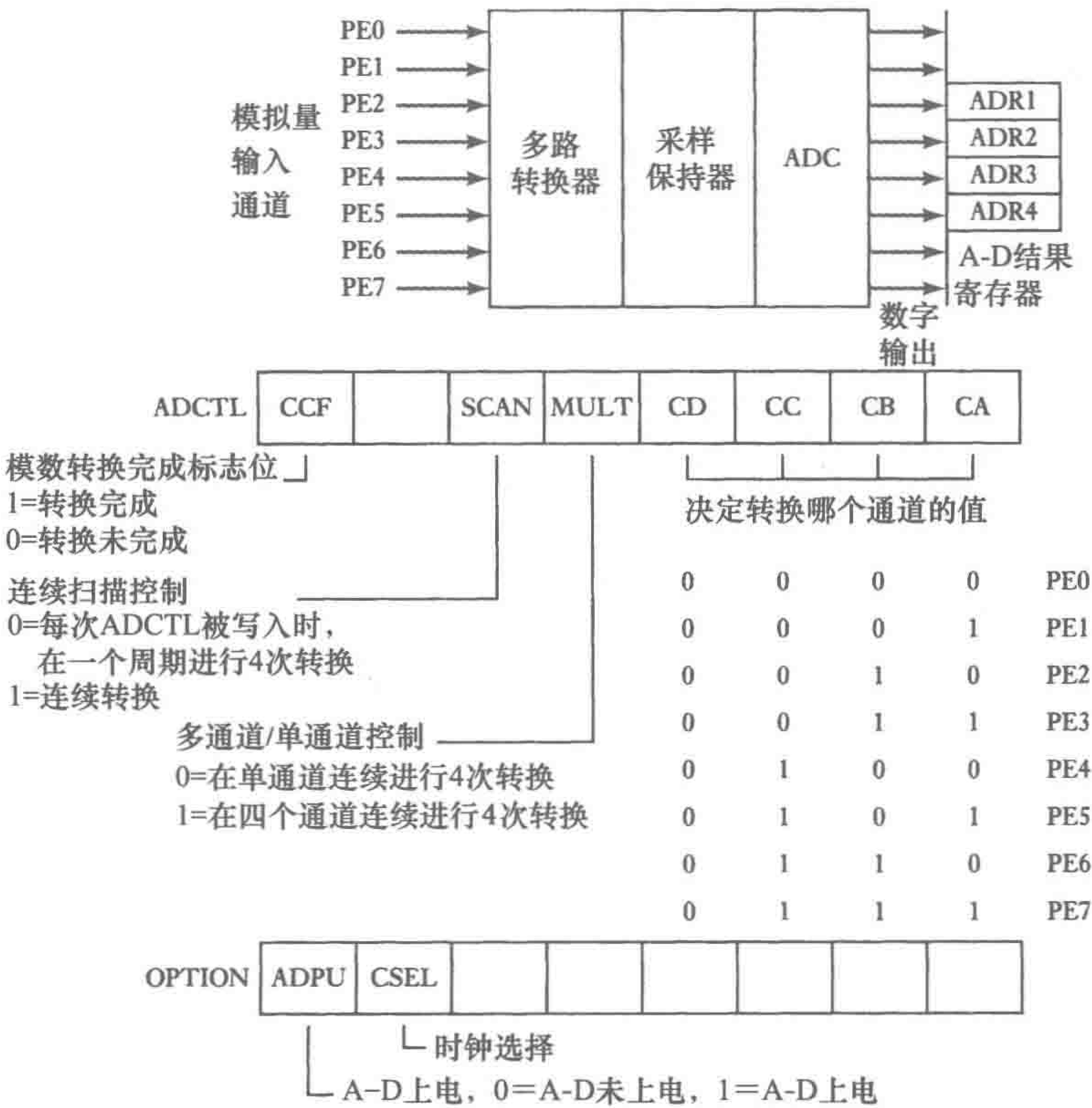


图 12-7　ADC

注意，| 是或运算符，结果只有在两个相应的位都为 0 时才为 0，否则为 1。这用于开启一个或多个位，或者设定为某一个值。在本例中，k=1，所以仅将 CA 设置为 1。可以导入延迟子程序，用来保证 ADC 启动后，值不会过快地被读取。

12.8　Arduino 编程

Sketch 是使用 Arduino 板编程时会用到的术语。Arduino 使用 C 语言进行编程，其程序的基本格式包含两个函数：setup 和 loop。setup 函数在程序刚开始时运行，用于引脚配置、声明变量和常数等。loop 函数通常逐步运行，当程序运行到循环底部时则自动返回到 loop 函数的第一步重新执行 loop 函数，直至程序停止。

```
void setup()
```

```
{
// set up 代码放在这里
}
void loop ()
{
// 一步步代码放在这里
}
```

值得注意的是，在 Arduino 编程中，第一个大括号 { 有时会放置在函数声明的后面，故以上程序就会呈现以下格式：

```
void setup() {
// set up 代码放在这里
}
void loop () {
// 一步步代码放在这里
}
```

setup 函数本身会调用两个内置函数：pinMode 和 digitalWrite。在 Arduino 编程中，其数字引脚既可作为输入又可作为输出，故 pinMode 函数会将一个特别的引脚设置为输入或输出。而 digitalWrite 函数则会将一个引脚设置成高或低两种电平状态。由于这两个函数均没有返回值，所以函数声明是空函数（void）。为了便于理解，我们以一个控制发光二极管（LED）的小程序为示例进行说明，该发光二极管的发光和熄灭由内连的引脚 13 控制。程序的注释可由//或/＊和＊/符号标注出，当注释内容仅有一行时可以使用//，当注释内容超过一行时则需要用/＊和＊/符号。被注释的内容不会作为代码内容进行编译并上传到微控制器。

320

```
// 打开内部 LED 0.5 s,然后关闭 0.5 s,重复上述过程
void setup ()
{
    pinMode (13, OUTPUT);
}
void loop ()
{
    digitalWrite (13, HIGH);
    delay (500);
    digital Write (13, LOW);
}
```

延时函数（delay）是为了在设置 13 号引脚的高低电平变化时加入 0.5s 的延时，即先将 13 号引脚标识为“高”（High），经过 0.5s 后修改该引脚标识为“低”（Low）。

假如我们想控制一个外接的 LED 亮 0.5s，熄灭 0.5s，如此反复。我们首先需要明确与该 LED 相连接的引脚，并将该引脚设置为一个输入。在连接 LED 时值得注意的一点是，在 20mA 的电流下，会在 LED 上产生 2V 的压降；而 Arduino 板提供的电压是 5V，故我们需要与 LED 串联一个电阻，使得电阻分担 3V 压降，从而使 LED 两端的电压刚好为 2V。串联电阻器的最小阻值为 $U/I = 3\text{V}/0.020\text{A} = 150\Omega$。不过由于 LED 在更小电流条件下也能发亮，所以我们通常会采用一个阻值高一些的电阻器。

```
// 关闭外部 LED 0.5 s,然后打开 0.5 s,重复上述过程
    #define ext_LED 12
void setup ()
{
    pinMode (ext_LED, OUTPUT);
```

```
}
void loop ()
{
    digitalWrite (ext_LED, LOW);
    delay (500);
    digitalWrite (ext_LED, HIGH);
    delay (500);
}
```

现在我们来考虑一下同时有一个内接和一个外接 LED 的情况。

321

```
/* 打开内部 LED 关闭外部 LED 0.5 s,然后关闭内部 LED 打开外部 LED 0.5 s,重复上述过程*/
#define int_LED 13
#define ext_LED 12
void setup ()
{
    pinMode (int_LED, OUTPUT);
    pinMode (ext_LED, OUTPUT);
}
void loop ()
{
    digitalWrite (int_LED, HIGH);
    digitalWrite (ext_LED, LOW);
    delay (500);
    digitalWrite (int_LED, LOW);
    digitalWrite (ext_LED, HIGH);
    delay (500);
}
```

以上的程序可以调控内接和外接的 LED 重复地交替闪烁，我们也可以在此基础上做出改进，使该交替性闪烁只能在一个开关闭合之后发生。

```
/* 如果一个开关闭合则打开内部 LED 和关闭外部 LED 0.5 s,然后关闭内部 LED 和打开外部 LED 0.5 s,重复上述过程*/
#define int_LED 13
#define ext_LED 12
#define ext_sw 11
Int switch_value;
void setup ()
{
    pinMode (int_LED, OUTPUT);
    pinMode (ext_LED, OUTPUT);
    pinMode (ext_sw. INPUT);
}
void loop ()
{
switch_value = digitalRead(ext_sw);
    if (switch_value ==LOW)
        {
        digitalWrite (int_LED, HIGH);
        digitalWrite (ext_LED, LOW);
        delay (500);
        digitalWrite (int_LED, LOW);
        digitalWrite (ext_LED, HIGH);
        delay (500);
        }
```

```
    else
        {
        digitalWrite(int_LED, LOW);
        digitalWrite(ext_LED, LOW)
        }
}
```

以上小程序仅是对 Arduino 编程的一个简单介绍，其中 Arduino 编程所需的关键过程基本已在本章前面用 C 语言展示出来，用户还可以在 Arduino 网站上找到一些已经编写好的程序包并免费使用。

为了使用 C 语言对 Arduino 进行编程，用户首先需要从 Arduino 网站上下载 Arduino Development Environment（Arduino 开发环境）。该程序可以使用用户直接通过 C 语言对 Arduino 进行编程，然后进行编译，并对用户写好的基于 C 语言规则的程序进行检查，将其翻译成汇编语言，再翻译成 Arduino 板能理解的机器码。当用户首次对 Arduino 板进行启动时，会进入 bootloader 程序。bootloader 程序是一段出厂时已经下载到内存中的代码，它可以使用用户通过 USB 接口上传程序。当 Arduino 板从用户电脑接收到上传程序的命令时，机器码程序会被加载到 Arduino 内存中，如此一来，Arduino 微控制器便可以在有需要时调用它。

以下是一系列基本操作：

1）从 Arduino 网站上下载 Arduino 开发环境到用户电脑。

2）通过一根 USB 电缆将 Arduino 板和主机相连。

3）启动 Arduino 开发环境。

4）在主机电脑上用 C 语言编写程序。

322 ~ 323

5）在屏幕上选中“上传”按钮。

6）程序在 Arduino 板上运行。

本章小结

C 语言是一种高级语言，较汇编语言更容易使用，同样的程序也可以用于不同的微处理器。所需要的就是选择合适的编译器，将 C 程序编译为相关微处理器的机器语言。汇编语言会因微处理器的不同而不同，但 C 语言却是标准化的。

C 语言程序包提供了库，库包括了大量的预先定义的函数。要想使用某一个库的内容，那个库就要在头文件里面声明。每一个 C 语言程序都必须有一个主函数 main()。当程序运行时，主函数起控制作用，并且也是第一个调用的函数。一个函数由多条语句组成，每一条语句都以一个分号结尾。语句也能用大括号 {} 括起。

习题

12.1 下面的问题都与程序的组成有关。

(a) 说出语句中什么被 int 定义：

```
int counter;
```

(b) 说出下面语句的作用：

```
num = 10;
```

(c) 说出下面语句的结果：

```
printf("Name");
```

(d) 说出下面语句的结果：

```
printf("Number % d", 12);
```

(e) 说出下面语句的作用：

```
#include <stdio.h>
```

12.2 对于下面的程序，说出其作用：(a) #include <stdio.h>；(b) {和}；(c) \n，并说出运行结果。

```
#include <stdio.h>
main( )
{
  printf("\n problem 3");
}
```

12.3 对于下面的程序，其输出结果是什么？

```
#include <stdio.h>

int main(void);
{
  int num;
  num = 20;

printf("The number is %d", num);
return 0;
}
```

12.4 写一个程序，从屏幕提示读取一个矩形的长和宽，计算其面积并输出结果，结果以"Area is"开头。

12.5 写一个程序，输出1~15，每行一个数。

12.6 下面的程序是输出两个数的商，解释各条语句的作用。

```
#include <stdio.h>

int main(void);
{
  int num1, num2;

  printf("Enter first number:");
  scanf("%d", &num1);

  printf("Enter second number: ");
  scanf("%d", &num2);

  if(num2 = = 0)
    printf("Cannot divide by zero")
  else
    printf("Answer is: %d", num1/num2);
return 0;
}
```

324
~
325

第13章　输入/输出系统

本章目的

通过本章的学习，读者将能够：

- 确定接口要求以及如何实现缓冲、握手、轮询和串行接口。
- 解释微控制器如何使用中断。
- 解释外设接口设备适配器的功能，并能针对具体环境对它进行编程。
- 解释异步通信接口适配器的功能。

13.1　接口

当使用微处理器控制某个系统时，它必须接收输入信息，并对之做出反应，同时产生输出信号来执行所需要的控制动作。所以系统中有传感器的输入信号来提供数据，有对继电器和电动机等外部设备的输出信号。**外设**这个术语就是用来形容连接到微处理器的外部设备，如传感器、键盘、驱动器等。但是，由于信号形式和电平高低缺乏兼容性，所以通常不能直接把外设连接到微处理器的总线系统上。针对这类不兼容性，所以一种称为接口的电路用在外设和微处理器之间。如图13-1所示，接口解决这种不兼容性。

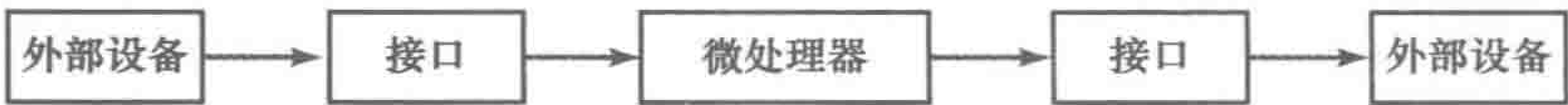


图13-1　接口

本章讨论这些接口的要求，以及广泛使用的摩托罗拉MC6820外设接口适配器和MC6850异步通信接口适配器。

13.2　输入/输出寻址

微处理器有两种方法选择输入/输出设备。有些微处理器，如Zilog的Z80，有**独立的输入/输出**，和专用的输入指令，如IN用来读取一个输入设备，以及专用输出指令，例如OUT用来输出到一个输出设备。例如，使用Z80，我们可以用：

326

```
IN A,(B2)
```

来读取B2设备的输入，并把数据放入累加器A中。输出指令可以是：

```
OUT(C),A
```

用来把累加器A中的数据写到端口C上。

更加普遍的情况是，微处理器没有独立的输入/输出指令，但与读/写内存使用相同的指令。这称为**内存映射的输入/输出**。使用这种方法，每个输入/输出设备都像内存单元一样有一个地址。摩托罗拉68HC11、Intel 8051和PIC微处理器都没有独立的输入/输出命令，它们都使用内存映射。因此，使用内存映射，我们用：

```
LDAA $1003
```

来读取地址为 $1003 的输入数据。使用：

```
STAA $1004
```

将输出数据写入地址 $1004。

微处理器使用并行端口来实现数据字节的输入和输出。许多的外设常需要多个输入/输出端口。这也许是外设的数据字比 CPU 的数据字长的原因。因此 CPU 必须将数据分段输出。例如，我们要用一个 8 位 CPU 产生一个 16 位的输出，过程如下：

1）CPU 准备这个数据的高 8 位。

2）CPU 把数据的高 8 位发送到第一个端口。

3）CPU 准备数据的低 8 位。

4）CPU 把数据的低 8 位发送到第二个端口。

5）因此，经过一定的延迟之后，所有的 16 位数据都提供给了外设。

13.2.1　输入/输出寄存器

摩托罗拉 68HC11 微控制器有 A、B、C、D、E 五个端口（见 10.3.1 节）。端口 A、C、D 是双向的，可用做输入或输出。端口 B 仅用做输出，而端口 E 仅用做输入。一个双向的端口是用做输入还是输出取决于控制寄存器中某位的设置。例如，处于地址 $1000 的端口 A 是由地址 $1026 的脉冲累加控制寄存器 PACTL 控制的。设置端口 A 为输入需将第 7 位置 0，设置为输出需将第 7 位置 1（见图 10-12）。端口 C 也是双向的，它的 8 位寄存器在地址 $1003，由其对应的位于地址 $1007 的方向控制寄存器控制。当对应的数据方向位设置 0，它就是输入；设置 1 时为输出。位于地址 $1008 的端口 D 是双向的，只含有 6 个输入/输出线。它被位于地址 $1009 的端口方向寄存器控制。每条线的方向是由控制寄存器的对应位来控制的。设置 0 为输入，设置 1 为输出。通过设置控制寄存器里的其他位，有些端口也可以被设置来实现其他功能。

327

对于一个固定方向的端口，例如，摩托罗拉 68HC11 的端口 B，仅作为输出，输出一些值的指令，例如 $FF，仅仅将值传递给地址。指令如下所示：

```
REGBAS  EQU    $1000      ;I/O 寄存器的基地址
PORTB   EQU    $04        ; PORTB 对 REGBAS 的偏移
        LDX    #REGBAS    ; 载入变址寄存器 X
        LDAA   # $FF      ; 向累加器加载 $FF
        STAA   PORTB,X    ; 将值存储到 PORTB 的地址
```

对于方向固定的端口 E，它仅用做输入使用，读取一个字节的指令如下所示：

```
REGBAS  EQU    $1000      ; I/O 寄存器的基地址
PORTE   EQU    $0A        ; PORTE 对 REGBAS 的偏移
        LDAA   PORTE,X    ; 将端口 E 中的值加载到累加器中
```

对于像端口 C 这样的双向端口，在把它用做输入端口前，我们需要配置才能把它用做输入端口。这意味着将所有的位都置 0。所以，我们可以这么做：

```
REGBAS  EQU    $1000      ;I/O 寄存器的基地址
PORTC   EQU    $03        ; PORTC 对 REGBAS 的偏移
DDRC    EQU    $07        ; 数据方向寄存器对 REGBAS 的偏移
        CLR    DDRC,X     ; 将 DDRC 全部位置 0
```

Intel 8051 微处理器（见 10.3.2 节）有四个并行的双向输入/输出端口。当一个端口位用做输出时，数据只是放入对应的特殊功能寄存器位；但它用做输入时，相关的位必须置 1，因此将整个端口写为 FFH。看看这个例子，当一个按钮按下时，Intel 8051 微控制器需要发出指令点亮一个 LED。这个按钮提供一个输入到 P3.1，一个输出到 P3.0。当按钮按下时，这个按钮输入低电平：

```
        SETB  P3.1     ; 将 P3.1 位置为 1,作为输入
LOOP    MOV   C,P3.1   ; 读取按钮的状态,将它存储到进位标志上
        CPL   C        ; 进位标志取反
        MOV   P3.0,C   ; 将进位标志复制到输出
        SJMP  LOOP     ; 重复以上事件序列
```

PIC 微控制器的双向端口的信号方向是由 TRIS 方向寄存器决定的（见 10.3.3 节）。TRIS 置 1 为读取，置 0 为写入。PIC16C73/74 的寄存器被安排为 2 个组（bank）。在选择一个特殊寄存器之前，需要设定 STATUS 寄存器的第 5 位来选择 bank。这个寄存器在两个 bank 中都有，不需要为了使用这个寄存器而选择 bank。TRIS 寄存器在 bank 1，PORT 寄存器在 bank 0。所以，为了把端口 B 作为输出，我们先选择 bank 1，将 TRISB 置 0。然后选择 bank 0，将输出写入到 PORTB。通过设定 STATUS 寄存器的一个位来选择 bank。将端口 B 作为输出的指令如下：

328

```
Output  clrf  PORTB       ; 将端口 B 清零
        bsf   STATUS,RP0  ; 将 RP0 置 1,用 STATUS 寄存器选择 bank 1
        clrf  TRISB       ; 清零输出位
        bcf   STATUS,RP0  ; 用 STATUS 寄存器选择 bank 0
                          ; 端口 B 为输出
```

13.3 接口要求

以下为接口电路常用的一些要求：

1. 电气缓冲/隔离

当外设工作的电压或电流与微处理器总线系统不匹配或有不同的参考接地时，电气缓冲/隔离是必要的。术语**缓冲器**（buffer）是指用于隔离和实现电流电压放大的设备。例如，如果微处理器的输出连接到晶体管的基极，晶体管通/断所需的基极电流大于微处理器所能提供的电流，这样一个缓冲器用于提升电流。在微处理器和较高的功率系统之间通常需要隔离。

2. 时序控制

当外设的数据传输速率与微处理器的不同时，例如在将微处理器连接到一个较慢的外设时，就需要时序控制。这可以通过在微处理器和外设之间使用专用线来控制数据传输的时序。这样的线称为**握手线**（handshake line），这样的过程称为**握手**（handshaking）。

3. 编码转换

当外设和微处理器所使用的编码不同时，这就需要编码转换。例如，LED 显示屏可能需要解码器，将从微处理器输出的 BCD 码转换成操作七段数码显示所需的代码。

4. 改变线数

微处理器的操作字长为固定的 4 位、8 位或 16 位。这就决定了在微处理器数据总线中线的数量。外设可能有不同数量的线路，也许需要比微处理器数据总线更长的字长。

5. 串行到并行、并行到串行的数据传输

在 8 位微处理器中，一般一次操作 8 位数据。要同时给外设传输 8 位数据，就需要 8 条数据线。这种传输的形式称为**并行数据传输**。然而，并不总是能够以这样的方式传输数据。例如，在公共电话系统的数据传输只可以涉及一条数据路径。因此，该数据必须按顺序一次 1 位地传输。这种传输的形式称为串行数据传输。串行数据传输的数据传输速度比并行数据传输慢。因此，如果使用串行数据传输，微处理器需要将输入的串行数据转换成并行数据，反之亦然。

329

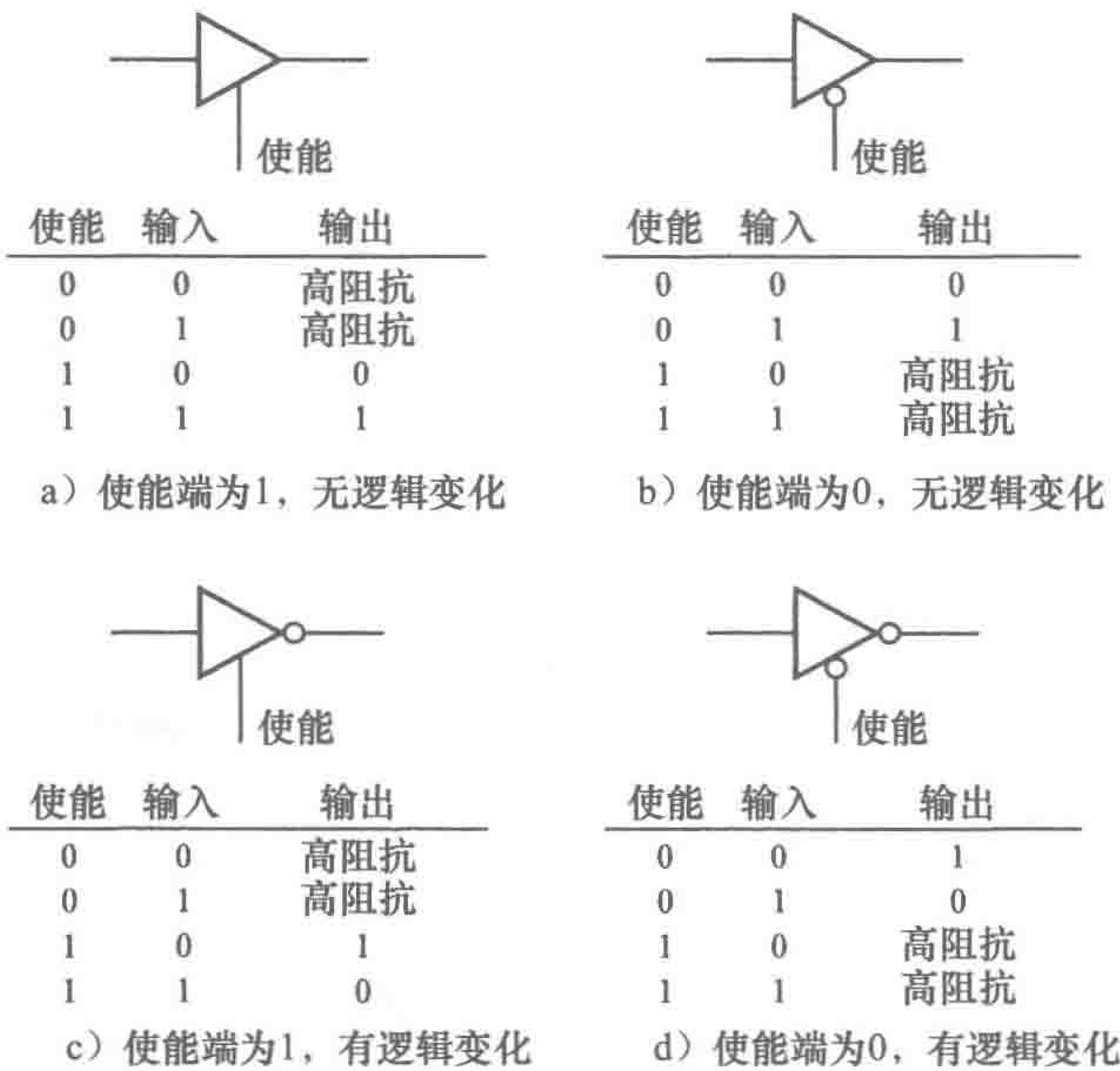
6. AD 和 DA 转换

传感器的输出通常是模拟量，这需要转换为数字信号给微处理器。微处理器的输出是数字量，这需要转换为模拟信号以控制某些驱动器。许多微控制器都具有内置的模数转换器，例如

PIC 16C74/74A（见图 10-30）和摩托罗拉的 M68HC11（见图 10-10），这样可以处理模拟输入。然而，在需要模拟输出时，微处理器一般要通过外部的数模转换器（例子见 13.6.2 节）。

13.3.1　缓冲器

缓冲器是一个连接系统的两个部分以防止干扰的设备。缓冲器的一个重要用途是在微处理器的输入端口中，用来隔离输入数据和微处理器数据总线，直到微处理器请求数据。常用的缓冲器是一个**三态缓冲器**。三态缓冲器由一个控制信号使能，提供逻辑 0 或 1 输出，未使能时它具有高阻抗，以便有效地切断电路。图 13-2 显示了三态缓冲器的符号和使能条件。如图 13-2a、b 所示的缓冲器不改变逻辑输入，图 13-2c、d 所示的则相反。



使能	输入	输出
0	0	高阻抗
0	1	高阻抗
1	0	0
1	1	1

a）使能端为1，无逻辑变化

使能	输入	输出
0	0	0
0	1	1
1	0	高阻抗
1	1	高阻抗

b）使能端为0，无逻辑变化

使能	输入	输出
0	0	高阻抗
0	1	高阻抗
1	0	1
1	1	0

c）使能端为1，有逻辑变化

使能	输入	输出
0	0	1
0	1	0
1	0	高阻抗
1	1	高阻抗

d）使能端为0，有逻辑变化

图 13-2　缓冲器

330

在 PIC 微控制器中（参见 10.3.3 节），TRIS 位连接到一个三态缓冲器的使能输入。如果该位为 0，则三态缓冲器使能，只是简单地将输入值传递到输出；如果该位为 1，则三态缓冲器被禁止，输出变为高阻抗（见图 13-2b）。

当某些外设需要与微处理器共享相同的数据线时就要使用这种三态缓冲器，即它们都连接到数据总线，因此需要微处理器只能一次使能一个设备，而其他不使能。图 13-3 显示了如何使用这些缓冲器。这些缓冲器是集成电路的，如 74125 有与四个非反相低电平有效缓冲器；74126 有四个非反相高电平有效的缓冲器。

13.3.2　握手

除非数据交换时两个设备可以用相同的速率发送和接收数据，否则握手是必要的。握手时，较慢的设备决定了数据传输速率。对于并行数据传输，**选通和应答**（strobe-and-acknowledge）是握手常用的形式。外设发送数据就绪（DATA READY）信号到输入/输出区。然后由 CPU 确认数据就绪信号是有效的。接着，CPU 读取数据输入/输出区的数据并发送输入应答（INPUT ACKNOWLEDGE）信号给外设。这个信号表明这次传送已经完成，因此外设可以发送更多的数据。对于输出，外设发送输出请求（OUTPUT REQUEST）或外设就绪（PERIPHERAL READY）信号给输入/输出区。CPU 确认外设就绪信号是有效的，就将数据发送到外设。下一个外设就绪信号通知 CPU 发送已完成。

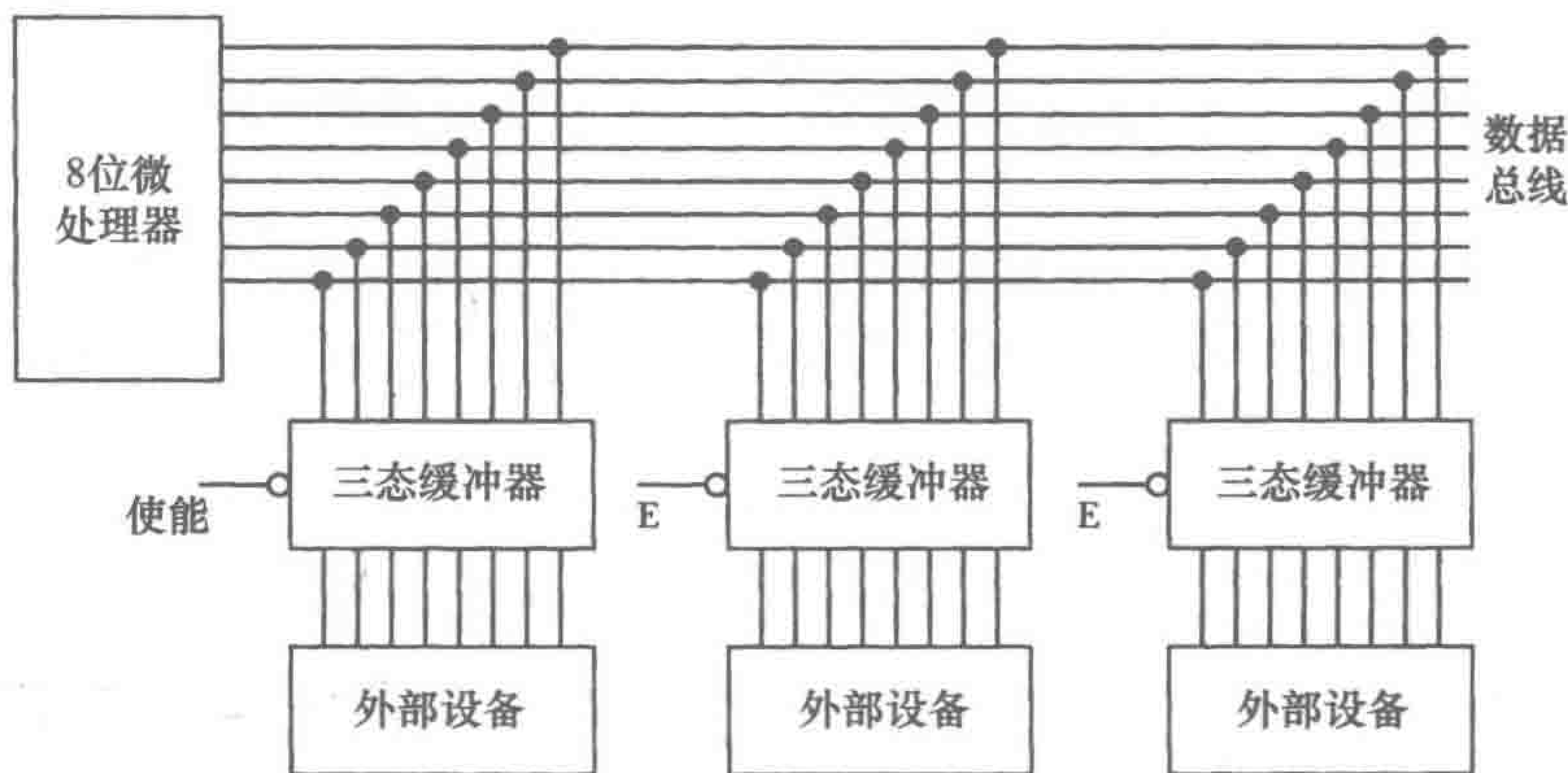


图 13-3　三态缓冲器



图 13-4　握手控制：选通和应答

使用微控制器 MC68HC11，基本的选通输入/输出操作如下。握手控制信号使用 STRA 和 STRB 引脚（如图 13-4a 所示，完整的框图模型见图 10-10），端口 C 用于选通输入，端口 B 用于选通输出。当微控制器已经准备好发送数据时，STRA 引脚产生一个脉冲，发送到外设。当微控制器在 STRB 引脚接收到一个上升沿或下降沿信号时，微控制器的相关输出端口将数据发送到外设。当数据已经准备好要传送到微控制器时，外设发送信号到 STRA 引脚表示准备就绪，然后 STRB 引脚上产生上升沿或下降沿来表示愿意接收。在握手发生前，必须首先配置地址为 $1002 的并行输入/输出寄存器 PIOC。图 13-4b 显示了该寄存器的相关位所需的状态。

完整的输入/输出握手（full handshake input/output）包括沿 STRB 引脚传送的两个信号，第一个表明准备接收数据，下一个表示已读取数据。这种操作模式需要把 PIOC 的 HNDS 位设置为 1。PLS 如果置 0，则完整的握手发送脉冲；若设置为 1，则为互锁。脉冲操作发送脉冲作为应答，STRB 互锁时复位（见图 13-5）。

13.3.3　轮询和中断

考虑如下所有数据的输入/输出都由程序控制的情况。当外设需要关注时，它们通过改变输入线路的电压电平来发送一个信号给微处理器。然后微处理器可以做出响应，跳转到为该设备设计的服务程序。在程序完成后，就返回到主程序。对输入/输出的控制程序就是一个连续读取输入和更新输出的循环，需要时跳转到服务程序。这种反复检查每一个外设，看它是否已经准备好发送或接收一个新的数据字节的过程称为**轮询**（polling）。

另一种程序控制是**中断控制**。中断包括外设启动一个单独的中断请求线。微处理器接收中断后，暂停正在执行的主程序，跳转到为外设设计的服务程序。中断必须不能导致数据丢失，中断处理程序必须包含在软件中，使处理器寄存器的状态和主程序访问的最后一个地址存储在

内存中的专用位置。中断服务程序结束后，内存中的内容恢复，微处理器可以从它被中断的地方继续执行主程序（如图 13-6 所示）。

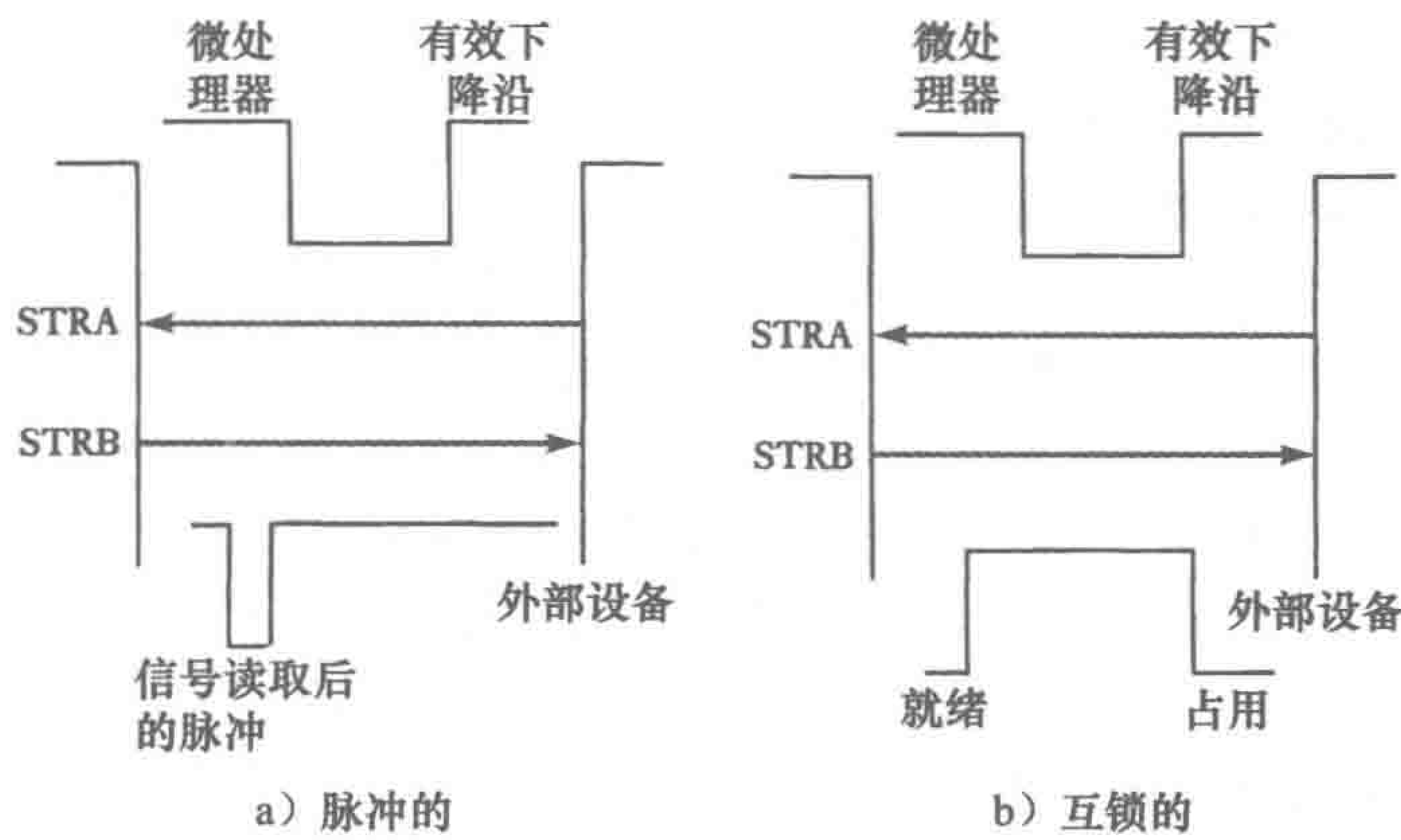


图 13-5 完整握手

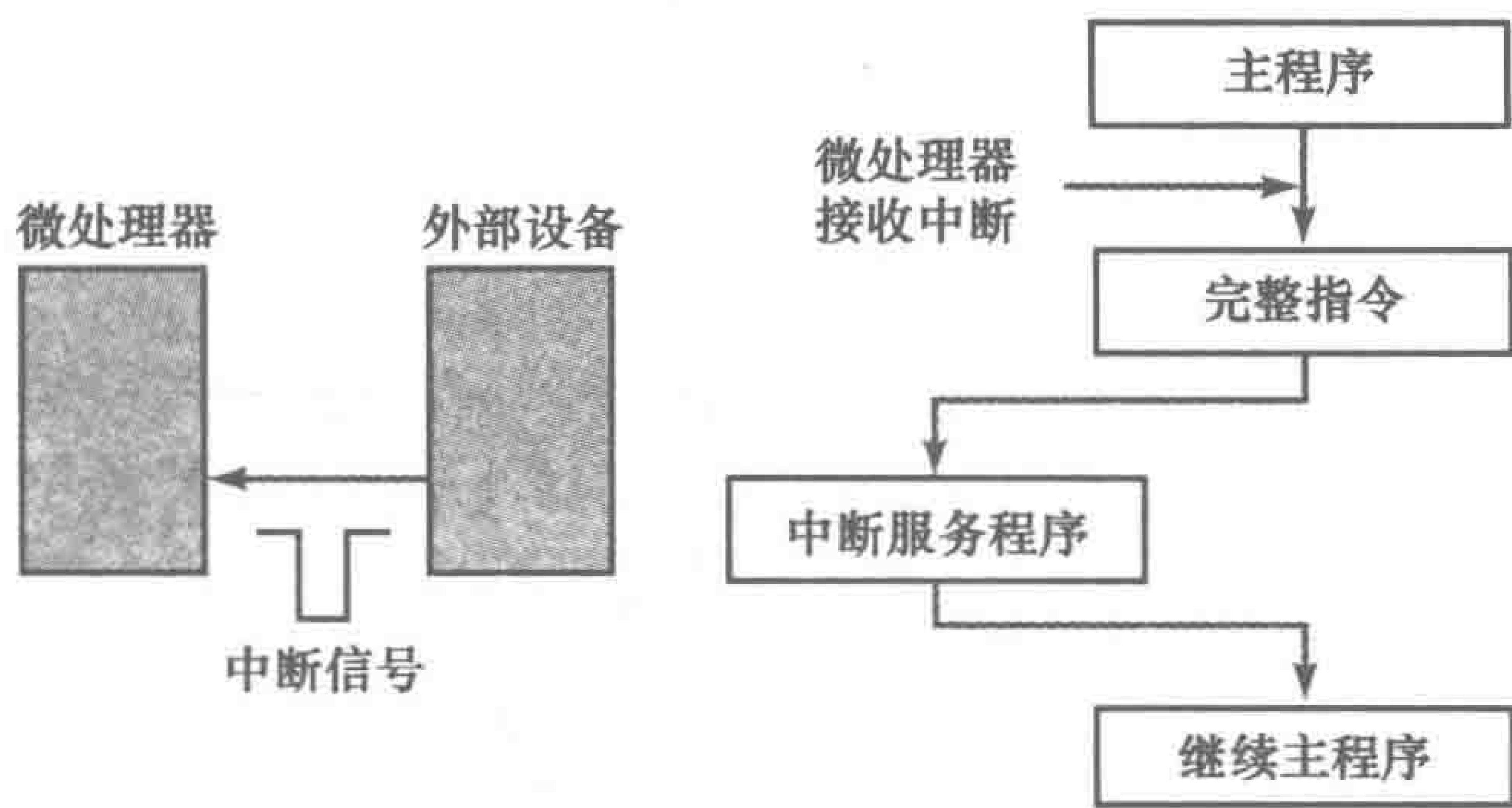


图 13-6 中断控制

因此，当中断发生时：

1）处理中断前，CPU 等待当前正在执行的指令结束。

2）所有的 CPU 寄存器入栈，设置一个标志位，防止此期间发生进一步中断。栈是内存中的一块特殊区域，当要执行一个子程序时，程序计数器值可以存储于其中。程序计数器提供程序中的下一个指令的地址，因此存储这个值就能使程序从执行中断的地方恢复。

3）然后，CPU 决定待执行的中断服务程序的地址。有些微处理器有专用的中断引脚，并由选定的引脚确定该地址。其他微处理器只有一个中断引脚，中断设备必须提供数据，告诉微处理器中断服务程序的入口地址。有些微处理器同时有这两种中断输入。中断服务程序的起始地址称为**中断向量**。分配存储这些向量的内存块称为**向量表**。向量地址是由芯片制造商确定的。

4）CPU 转向中断服务程序。

5）这个程序完成后，栈返回 CPU 寄存器，主程序从中断的地方继续执行。

331
~
333

与位于程序中特定点的子程序调用不同，中断可以在程序中的任何点被调用。需要注意的是程序并不控制中断何时产生，而取决于中断事件。

因为一般硬件不能等待，所以输入/输出操作经常使用中断。例如，当一个键被按下时，键盘会产生一个中断输入信号。然后，微处理器挂起主程序，处理从键盘输入的信息。处理完成后，返回到主程序，继续从中断的地方执行。这种将任务编写为中断服务程序，并把它与一

个外部信号联系在一起的能力简化了许多控制任务，使其能够毫不拖延地处理。对于某些中断，可以对微处理器编程，使之忽略中断请求信号，除非使能位已置位。这称为**中断屏蔽**。

摩托罗拉 68HC11 有两个外部中断请求输入。XIRQ 是一个非屏蔽中断，总是会在当前正在执行的指令完成时执行。当 XIRQ 中断发生时，CPU 会跳转到中断服务程序，其中断向量地址存储在 $FFF4/5（地址的低字节和高字节）。IRQ 是一个可屏蔽中断。当微处理器的中断请求引脚 IRQ 接收到低电平信号时，微控制器跳转到由中断向量 $FFF2/3 指示的中断服务程序。IRQ 可通过设置中断屏蔽指令 SEI 屏蔽并通过清除中断屏蔽指令 CLI 清除屏蔽。在中断服务程序结束时，指令 RTI 用于返回到主程序。

对于 Intel 8051，中断源是通过位于地址 0A8H 的位寻址寄存器 IE（中断使能）来单独使能或禁止的（见图 10-26），0 禁止中断，1 使能中断。此外，在 IE 寄存器中还有一个全局的使能/禁止位，置位以使能所有的外部中断或清零以禁止所有外部中断。TCON 寄存器（见图 10-25）用于确定中断输入信号的类型，并启动一个中断。

对于 PIC 微控制器，中断由 INTCON 寄存器控制（见图 13-7）。为了使用端口 B 的第 0 位中断，它必须设置为输入，而且 INTCON 寄存器必须初始化使 INTE 为 1，GIE 为 1。如果中断发生在上升沿，那么 OPTION 寄存器中的 INTEDG 位（第 6 位）必须置 1（见图 10-32）；如果发生在下降沿，它必须置为 0。当中断发生时，INTF 被置位。它可以用指令 bcf INTCON，INTF 清除。

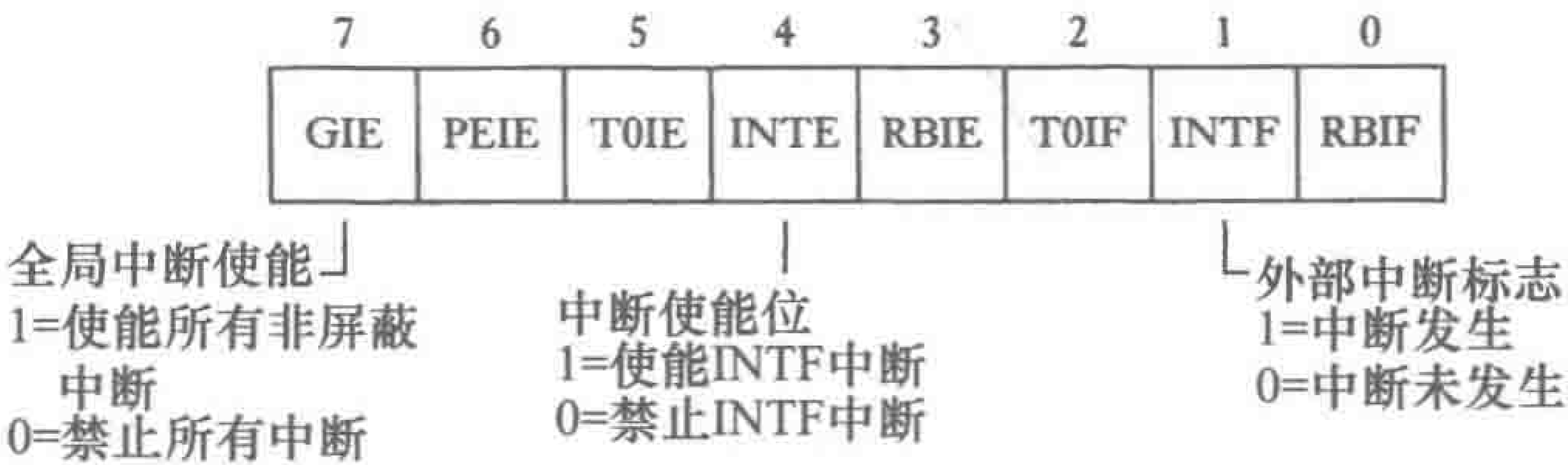


图 13-7　INTCON

以一个使用外部中断的程序为例，考虑一套简单的开/关控制程序，它是基于 Intel 8051 微控制器的中央供暖系统（见图 13-8）。P1.7 的输出控制中央加热炉，它使用了两个温度传感器，一个用来确定温度低于 20.5℃，另一个确定温度上升到 21.0℃。监测 21.0℃ 的温度传感器连接中断 INT0，端口 P3.2；监测 20.5℃ 的温度传感器连接到中断 INT1，端口 P3.3。将 TCON 寄存器的 IT1 位设置为 1，外部中断设置为边沿触发，即有一个从 1 到 0 的跳变就会触发。当温度升高到 21.0℃ 时，外部中断 INT0 的输入从 1 到 0 变化，中断被触发，执行 CLR P1.7 指令，将它清零，关闭加热器。当温度下降到 20.5℃ 时，外部中断 INT1 的输入从 1 到 0 变化，中断被触发，执行 SETB P1.7 指令，将它置 1 输出，打开加热器。主程序只是一组配置和使能中断的指令，建立初始条件来控制加热炉，即如果温度低于 20.5℃ 时，则打开；高于 21.0℃ 时，则关闭。然后什么也不做地等待中断发生。使用下面的程序，假设已有头文件，则：

334

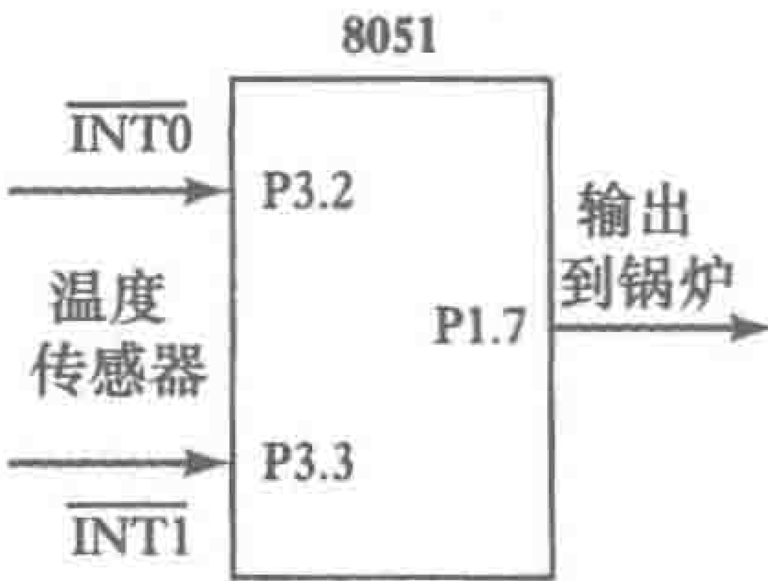


图 13-8　中央供暖系统

```
        ORG 0
        LJMP        MAIN
        ORG         0003H           ；给出 ISR0 的入口地址
ISR0    CLR         P1.7            ；关闭加热器的中断服务程序
        RETI                        ；中断返回
        ORG         0013H           ；给出 ISR1 的入口地址
ISR1    SETB        1.7             ；打开加热器的中断服务程序
        RETI                        ；中断返回
```

```
        ORG     30H
MAIN    SETB    EX0         ; 使能 INT0
        SETB    EX1         ; 使能 INT1
        SETB    IT0         ; 设置下降沿触发
        SETB    IT1         ; 设置下降沿触发
        SETB    P1.7        ; 打开加热器
        JB      P3.2,HERE   ; 如果 P3.2 等于 1,跳转到 HERE,加热器仍然打开
        CLR     P1.7        ; 关闭加热器
HERE    SJMP    HERE        ; 等待,直到发生中断
        END
```

除了中断请求外，微控制器还有复位中断和不可屏蔽中断。复位中断是一个特殊类型的中断，当触发时系统复位。因此它被触发时，系统的所有活动停止，加载主程序的起始地址和执行启动程序。微控制器 M68HC11 有一个计算机操作正常（COP）看门狗定时器。当在规定的时间内 CPU 不执行代码的某些部分，就会检测到软件处理错误。当这种情况发生时，COP 定时器超时，系统复位启动。

非屏蔽中断不能被屏蔽，所以当它连接到线上时，没有方法阻止中断服务程序的执行。这种类型的中断通常是保留给那些类似有电源故障时所需的紧急程序，例如切换到后备电源。 335

13.3.4 串行接口

对于并行数据传输，一条线用于一个位；而串行系统，使用一根线根据位的顺序进行数据传输。串行数据传输有两个基本类型：异步和同步。

对于**异步传输**，接收器和发送器都使用自己的时钟信号，所以接收器不可能知道一个字何时开始或停止。因此，有必要为每个传输的数据字设置自身的启动和停止位，这样接收器可以知道何处一个字停止和另一个字开始（见图 13-9）。对于这样的传输模式，发送器和接收器的距离一般很远（标准接口的详细信息，请参阅第 15 章）。对于**同步传输**，发送器和接收器有一个共同的时钟信号，从而可以同步传输和接收。

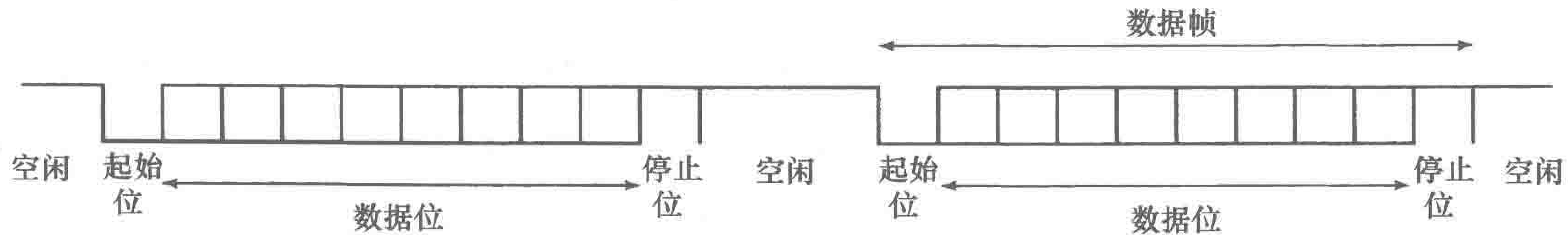


图 13-9 异步传输

微控制器 MC68HC11（见图 10-10）有一个串行通信接口（SCI）用于异步传输，因此可用于远程外设通信。SCI 使用端口 D 的引脚 PD1 作为发送线，引脚 PD0 作为接收线。这些线可以通过 SCI 控制寄存器使能或禁止。该微控制器还具有用来同步传输的串行外设接口（SPI）。这可用于局部串行通信，所谓局部实际上就是指位于芯片所在的机器中。

13.4 外设接口适配器

我们可以为特定的输入/输出专门设计接口。然而，也有可编程输入/输出接口设备，可通过软件选择各种不同的输入和输出选项。这种设备称为**外设接口适配器**（PIA）。

一个常用的 PIA 并行接口是摩托罗拉 MC6821。它是 MC6800 系列的一部分，可以直接连接到摩托罗拉 MC6800 和 MC68HC11 总线上。该设备可以认为是有两个带控制逻辑的并行输入/输出端口，实现与主机微处理器的连接。图 13-10 是 MC6821 外设接口适配器的基本结构和其引脚连接。

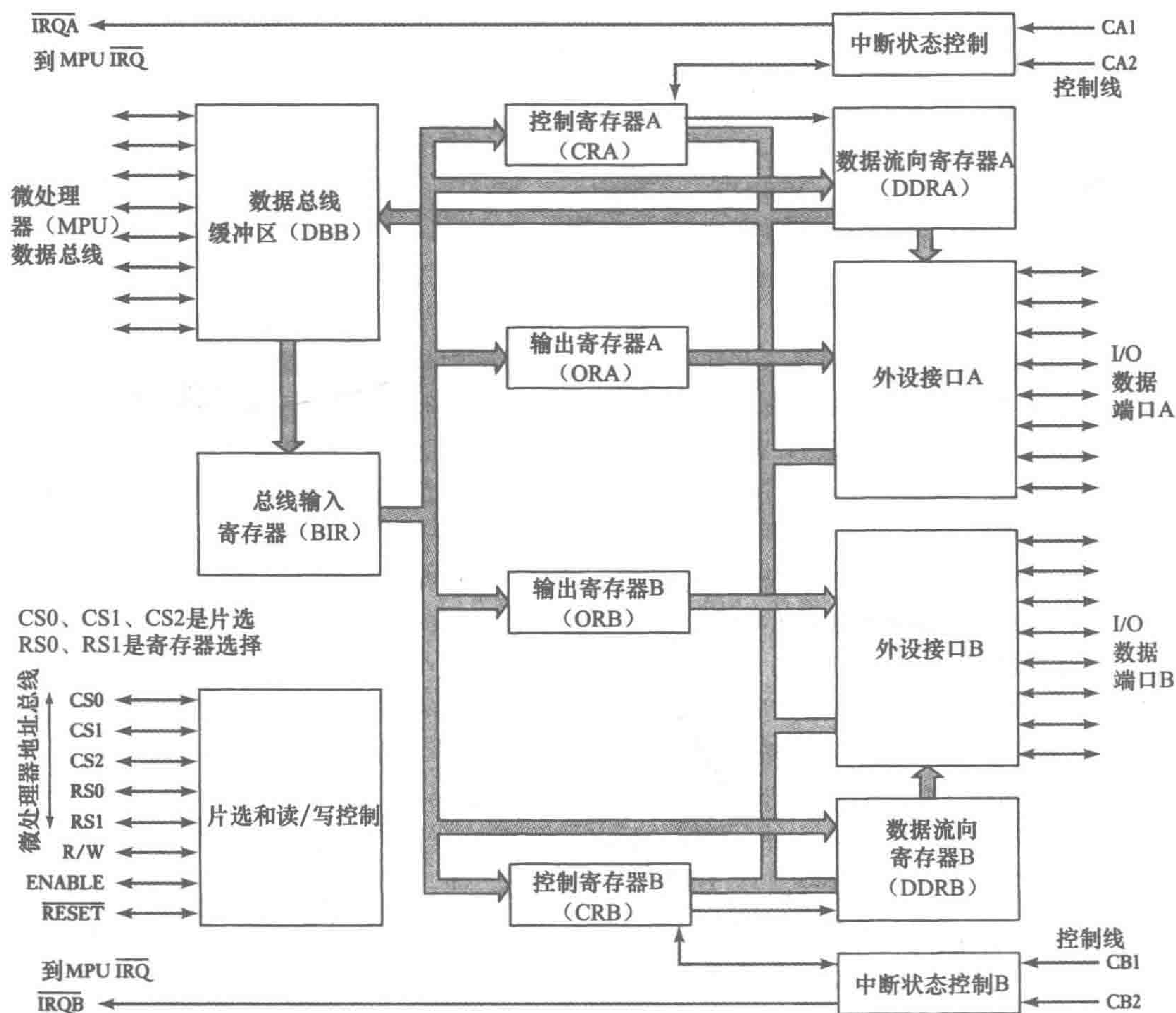


图 13-10　MC6821 外部接口适配器

这个外设接口适配器包含两个 8 位的并行数据端口，称为 A 和 B。每个端口有：

1）外设接口寄存器。因为必须为外设保存数据，所以输出端口工作方式与输入端口不同。因此，输出寄存器用来暂时存储数据。当用于输出端口时寄存器称为**被锁存连接的寄存器**，而当用于输入时，称为非锁存寄存器。

2）数据方向寄存器，用来决定输入/输出线是输入还是输出。

3）控制寄存器，决定外设中有效的逻辑连接。

4）两根控制线，CA1 和 CA2 或 CB1 和 CB2。

两个微处理器的地址线直接通过两个寄存器选择线 RS0 和 RS1 连接到 PIA。对于 6 个寄存器，这里给出了 PIA 4 个地址。当 RS1 为低电平时，A 被寻址；当它为高电平时，B 被寻址。RS0 将寄存器编址在特定的一侧，即 A 或 B。RS0 为高电平时，控制寄存器被寻址，为低电平时，数据寄存器或数据方向寄存器被寻址。对于特定的一侧，数据寄存器和数据方向寄存器有相同的地址。哪个寄存器被寻址是通过控制寄存器的第 2 位来决定的（见后面的论述）。

A 和 B 控制寄存器的每一个位都关系着端口的操作特征。A 控制寄存器的每一位如图 13-11 所示。B 控制寄存器使用类似的模式。

B7	B6	B5	B4	B3	B2	B1	B0
IRQA1	IRQA2	CA2控制			DDRA 存取	CA1控制	

图 13-11　控制寄存器

336
~
337

1. 位 0 和位 1

前两位控制 CA1 或 CB1 输入控制线的操作方式。位 0 决定是否使能中断输出。B0 =0 则禁止微处理器中断 IRQA(B)，B0 =1 则使能中断。CA1 和 CB1 不是由静态电平设置的，而是由边沿信号触发的，也就是一个改变的信号。位 1 决定位 7 是由一个从高到低的转变（下降沿）还是一个从低到高的转变（上升沿）设置。B1 =0 设置为下降沿，B1 =1 设置为上升沿。

2. 位 2

位 2 决定是否数据方向寄存器或外设数据寄存器被寻址。B2 设置为 0，则数据方向寄存器被寻址；B2 设置为 1，则外设数据寄存器被选中。

3. 位 3、位 4、位 5

这些位允许 PIA 执行多种功能。位 5 确定控制线 2 是输入还是输出。如果位 5 设置为 0，则控制线 2 为输入；如果设置为 1，则它是一个输出线。在输入模式下，CA2 和 CB2 都以相同的方式工作。位 3 和位 4 确定中断输出是否有效和位 6 的边沿转变方式设置。

当 B5 =0，即 CA2(CB2) 设置为输入时：B3 =0，则禁止由 CA2(CB2) 触发的微处理器中断 IRQA(B)；B3 =1 使能由 CA2(CB2) 触发的微处理器中断 IRQA(B)；B4 =0 决定中断标志位 IRQA(B)，即位6，是由 CA2(CB2) 设置为下降沿；B4 =1 则设置为上升沿。

B5 =1 将 CA2(CB2) 设置为输出。在输出模式，CA2 和 CB2 的表现不同。对于 CA2：当 B4 =0 和 B3 =0 时，使能（E）产生一个下降沿，CA2 变为低电平，微处理器读外设数据寄存器 A 的数据。随后等 CA1 发生变化后则返回高电平；当 B4 =0 和 B3 =1 时，使能（E）产生一个下降沿，CA2 变为低电平，微处理器读外设数据寄存器 A 的数据。在下一个下降沿使能后，则返回高电平。对于 CB2：当 B4 =0 和 B3 =0 时，使能（E）产生一个上升沿，CB2 变为低电平，微处理器写外设数据寄存器 B。下一个 CB1 转换后变回高电平。当 B4 =0 和 B3 =1 时，使能（E）产生一个上升沿，CB2 变为低电平，微处理器写外设数据寄存器 B。在下一个上升沿使能后则返回高电平。当 B4 =1 和 B3 =0 时，微处理器将 B3 =0 写入控制寄存器的同时，CA2(CB2) 变为低电平。当 B4 =0 和 B3 =1 时，CA2(CB2) 变为高电平，同时微处理器将 B3 =1 写入控制寄存器。

4. 位 6

这是 CA2(CB2) 的中断标志位，由 CA2(CB2) 上的过渡沿设置。当 CA2(CB2) 作为输入时（B5 =0），通过微处理器读数据寄存器 A(B) 清零。当 CA2(CB2) 作为输出时（B5 =1），标志为 0，不会被 CA2(CB2) 变化所影响。

5. 位 7

这是 CA1(CB1) 的中断标志位，微处理器读数据寄存器 A(B) 后清零。

设置要使用的选项的过程称为**配置**或**初始化** PIA。RESET 用于清除 PIA 的所有寄存器。然后 PIA 必须被初始化。

338

13.4.1 PIA 初始化

在可以使用 PIA 前，必须编写一个程序，设置期望的外设数据流的环境。PIA 程序放置在主程序的开始，所以微处理器可以读取外设数据。而初始化程序只需要运行一次。

设置端口是输入还是输出的初始化程序的步骤如下：

1）通过复位清除每个控制寄存器的位 2，因此寻址数据方向寄存器。数据方向寄存器 A 以 XXX0 方式寻址，而数据方向寄存器 B 以 XXX2 方式寻址。

2）A 作为输入端口，将方向寄存器 A 全写为 0。

3）B 作为输出端口，将方向寄存器 B 全写为 1。

4）将两个控制寄存器的位 2 写为 1。数据寄存器 A 以 XXX0 方式寻址，而数据寄存器 B

以 XXX2 方式寻址。

因此，复位后，一个用汇编语言编写的使 A 侧为输入和 B 侧为输出的初始化程序如下：

```
INIT LDAA  #$00  ;载入0
     STAA  $2000 ;A侧为输入端口
     LDAA  #$FF  ;载入1
     STAA  $2000 ;B侧为输出端口
     LDAA  #$04  ;位2设为1,其他位都是0
     STAA  $2000 ;选择端口A数据寄存器
     STAA  $2002 ;选择端口B数据寄存器
```

现在可以使用指令 LDAA 2000 从输入端口 A 读取外设数据，微处理器可以使用指令 STAA 2002 将外设数据写入输出端口 B。

13.4.2　通过 PIA 连接中断信号

摩托罗拉 MC6821 PIA（见图 13-12）有两个连接 IRQA 和 IRQB，通过它们中断信号可以传送到微处理器，使来自 CA1、CA2 或 CB1、CB2 的中断请求可以驱动微处理器的 IRQ 引脚为低电平有效状态。用上一节的思路来初始化 PIA，其中只有控制寄存器的位 2 设置为 1，其他位设置为 0。这些零禁止中断输入。为了使用中断，向控制寄存器写入 $04 的初始化步骤必须进行修改。修改的形式将取决于初始化中断所需输入类型的变化。

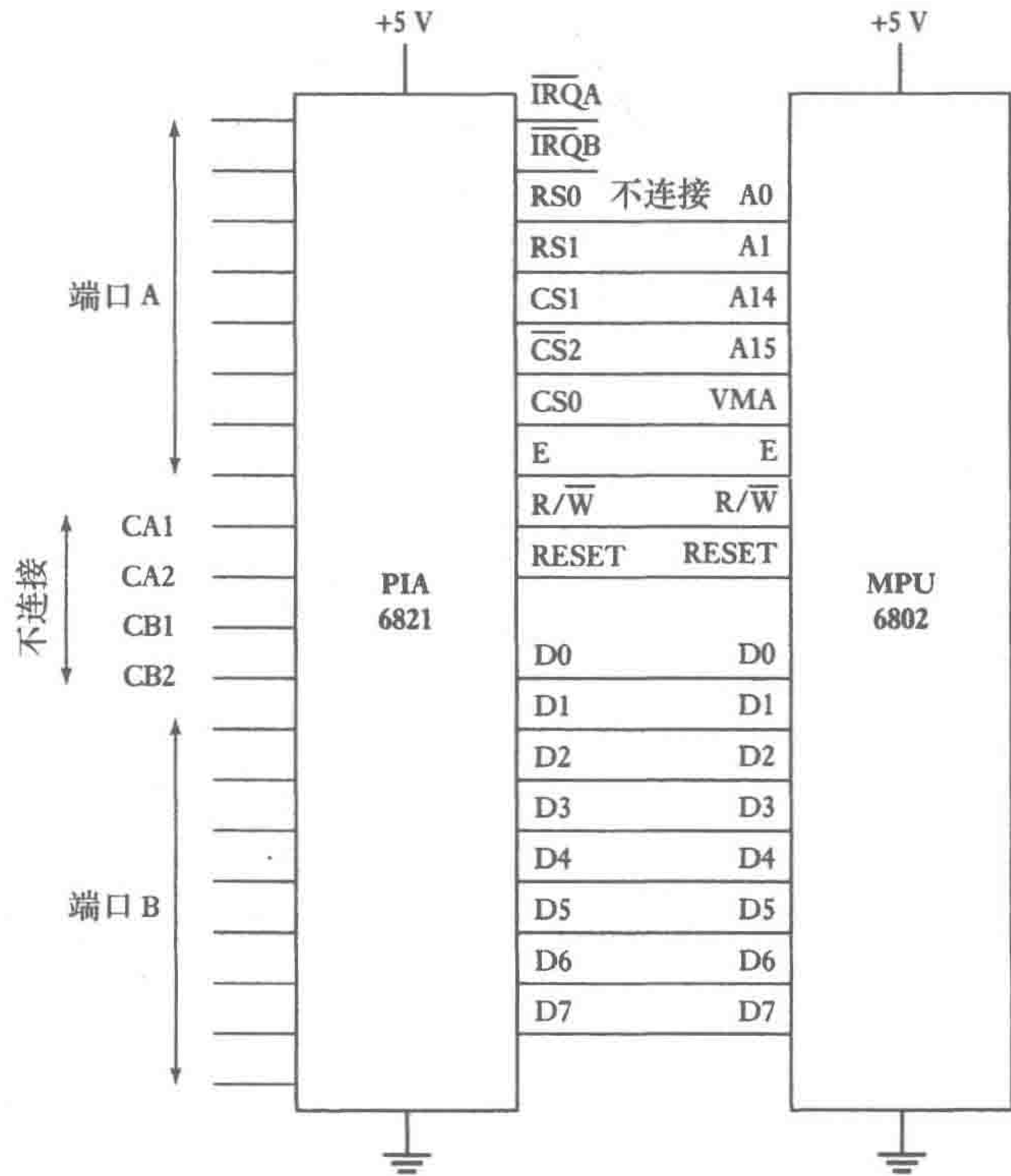


图 13-12　与 PIA 的接口

例如，假设我们想要 CA1 使能一个下降沿触发的中断，不使用 CA2 和 CB1，而用 CB2 作为一个置位/复位输出。对于 CA，满足本要求的控制寄存器格式如下：

B0 置 1，使能 CA1 上的中断。

B1 置 0，使得 CA1 上的中断标志位 IRQA1 为下降沿触发。

B2 置 1，可以访问数据寄存器。

B3、B4、B5 置 0,因为 CA2 未使能。
B6、B7 为只读标志位,0 或 1 可以被使用。

因此 CA1 的格式可以是 00000101，以十六进制表示为 05。CB2 控制寄存器的格式是：

B0 置 0,禁用 CB1。
B1 可以是 0 或 1,因为 CB1 被禁用。
B2 置 1,可以访问数据寄存器。
B3 置 0,B4、B5 置 1,选择置位/复位。
B6、B7 为只读标志位,0 或 1 可以被使用。

因此 CB2 的格式可以是 00110100，以十六进制表示为 34。初始化程序如下：

```
INIT  LDAA  #$00   ; 载入 0
      STAA  $2000  ; A 侧为输入端口
      LDAA  #$FF   ; 载入 1
      STAA  $2000  ; B 侧为输出端口
      LDAA  #$05   ; 载入所需的控制寄存器格式
      STAA  $2000  ; 选择端口 A 数据寄存器
      LDAA  #$34   ; 载入所需的控制寄存器格式
      STAA  $2002  ; 选择端口 B 数据寄存器
```

339~340

13.4.3 与 PIA 接口的例子

作为 PIA 接口的例子，图 13-13 为一个用于单极步进电动机的电路（见 9.7.2 节）。当开关打开时，电感绕组可以产生很强的反电动势，所以需要以某种方式把 PIA 与绕组隔离。可以使用光隔离器、二极管或电阻。二极管提供廉价和简单的接口，电阻不能完全隔离 PIA。

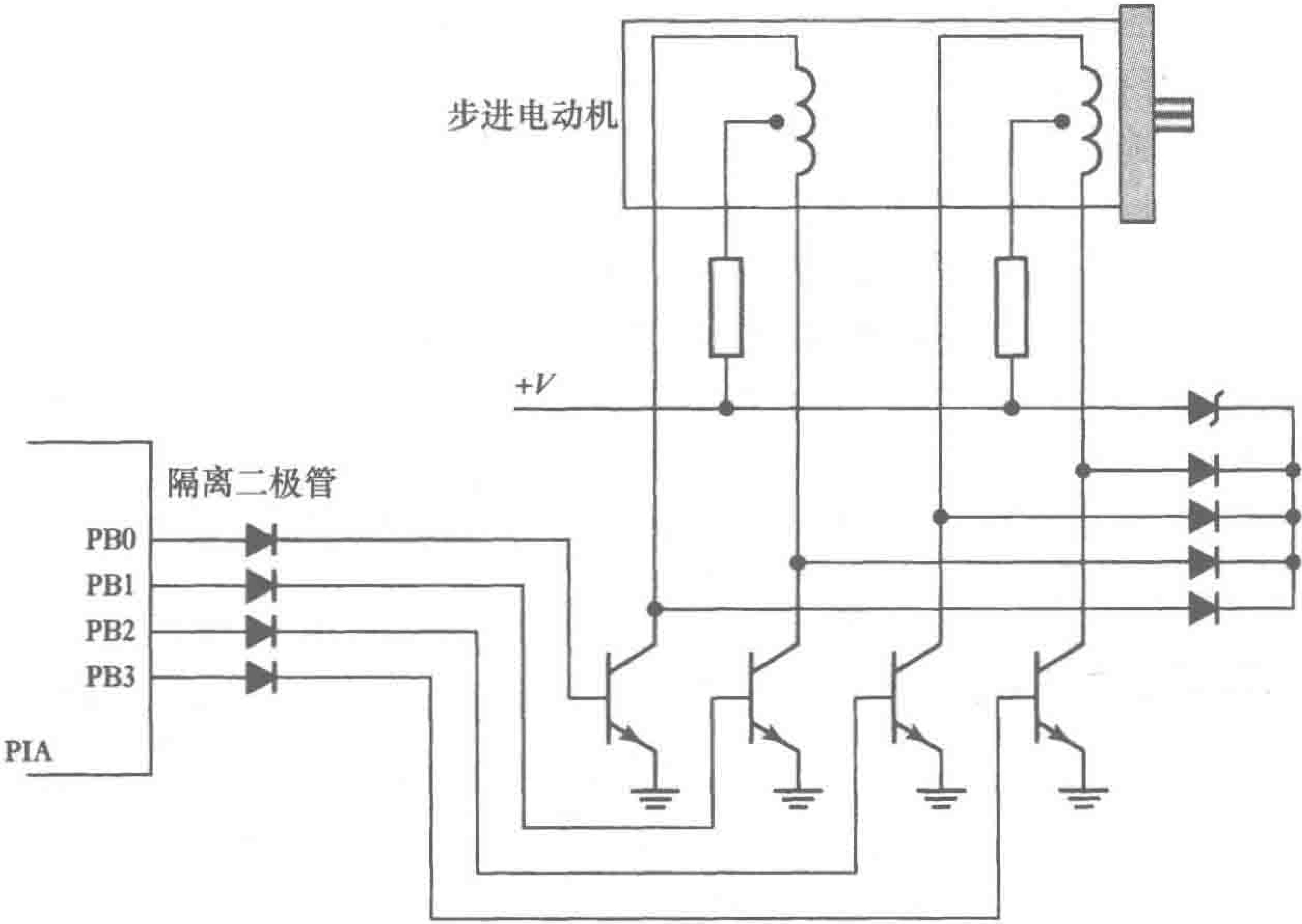


图 13-13 步进电动机接口

13.5 串行通信接口

通用异步接收/发送器（UART）是串行通信系统的基本组成元素，它的功能是将串行数据转换为并行数据作为输入和将并行数据转换为串行数据输出。一种常见的可编程形式 UART 是摩托罗拉 MC6850 **异步通信接口适配器**（ACIA）。图 13-14 为其构成要素的框图。

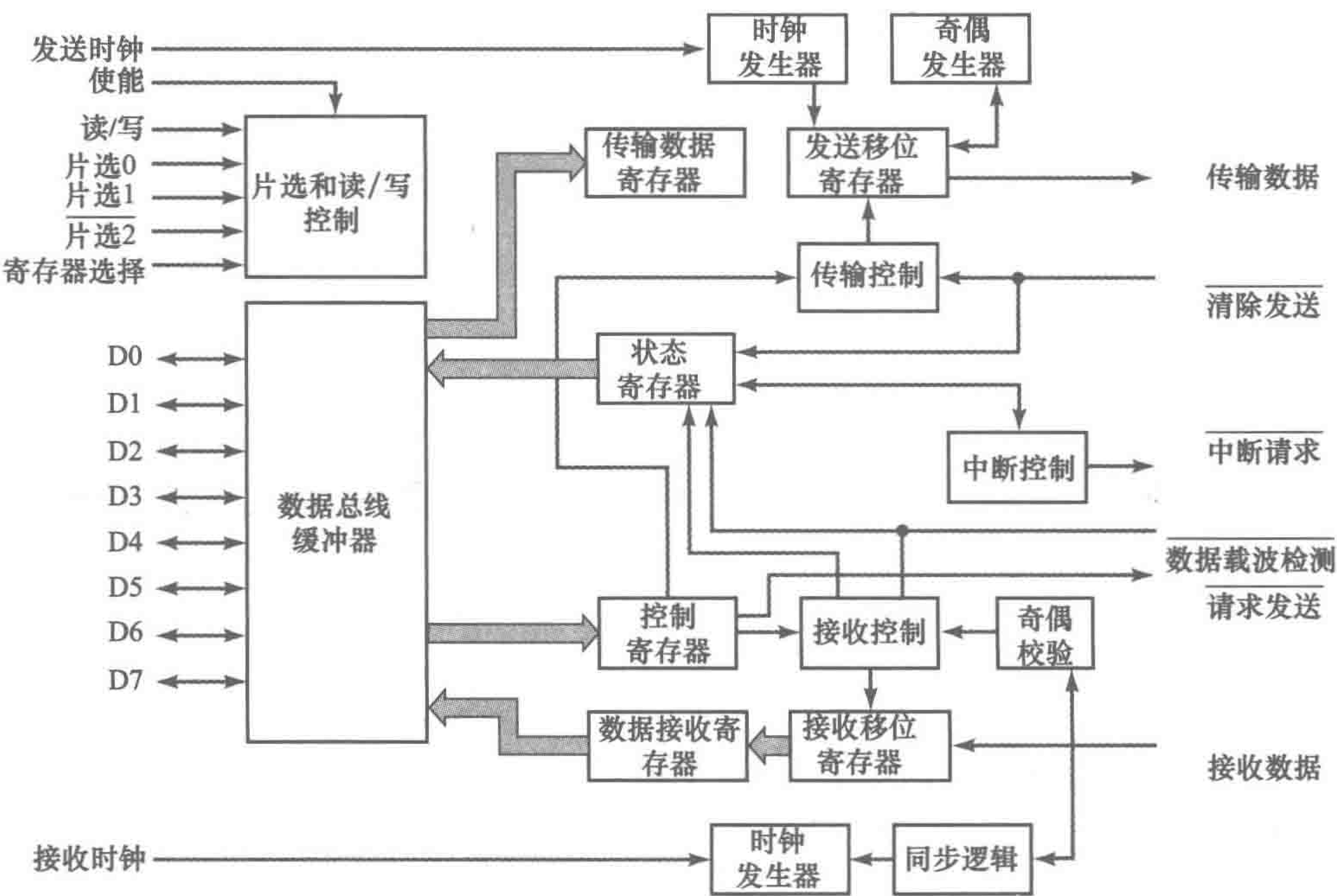


图 13-14　MC6850 ACIA

微处理器与 ACIA 之间的数据传输是通过 8 根双向线 D0 ~ D7 来实现的。微处理器通过对 ACIA 的读/写输入来控制数据流的方向。3 根片选线用于为 ACIA 分配地址。寄存器选择线用于 ACIA 选择特定的寄存器。如果寄存器选择线为高电平，则数据传输和数据寄存器被选中，如果为低电平，则控制和状态寄存器被选中。状态寄存器包含串行数据传输的状态信息，用于读取数据载波检测（data carrier detect）和清除发送（clear to send）线。控制寄存器最初用于复位 ACIA，后来用于定义串行数据传输速率和数据的格式。

ACIA 的外设侧包括两根串行数据线和 3 根控制线。数据由数据发送线发送并由数据接收线接收。控制信号由清除发送、数据载波检测和请求发送产生。图 13-15 为控制的位格式，图 13-16为状态寄存器。

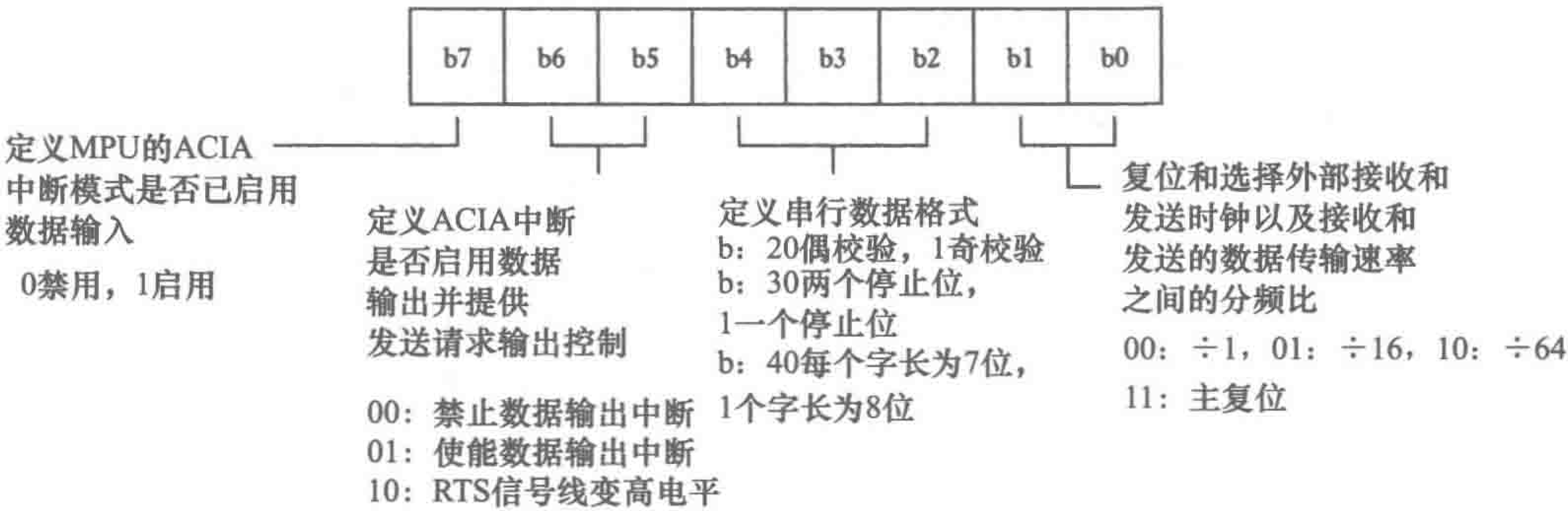


图 13-15　控制寄存器

341 ~ 342

异步串行数据传输一般用于两台计算机之间的通信（有或没有调制解调器）或者一台计算机与打印机之间的通信（更多讨论见第 15 章）。

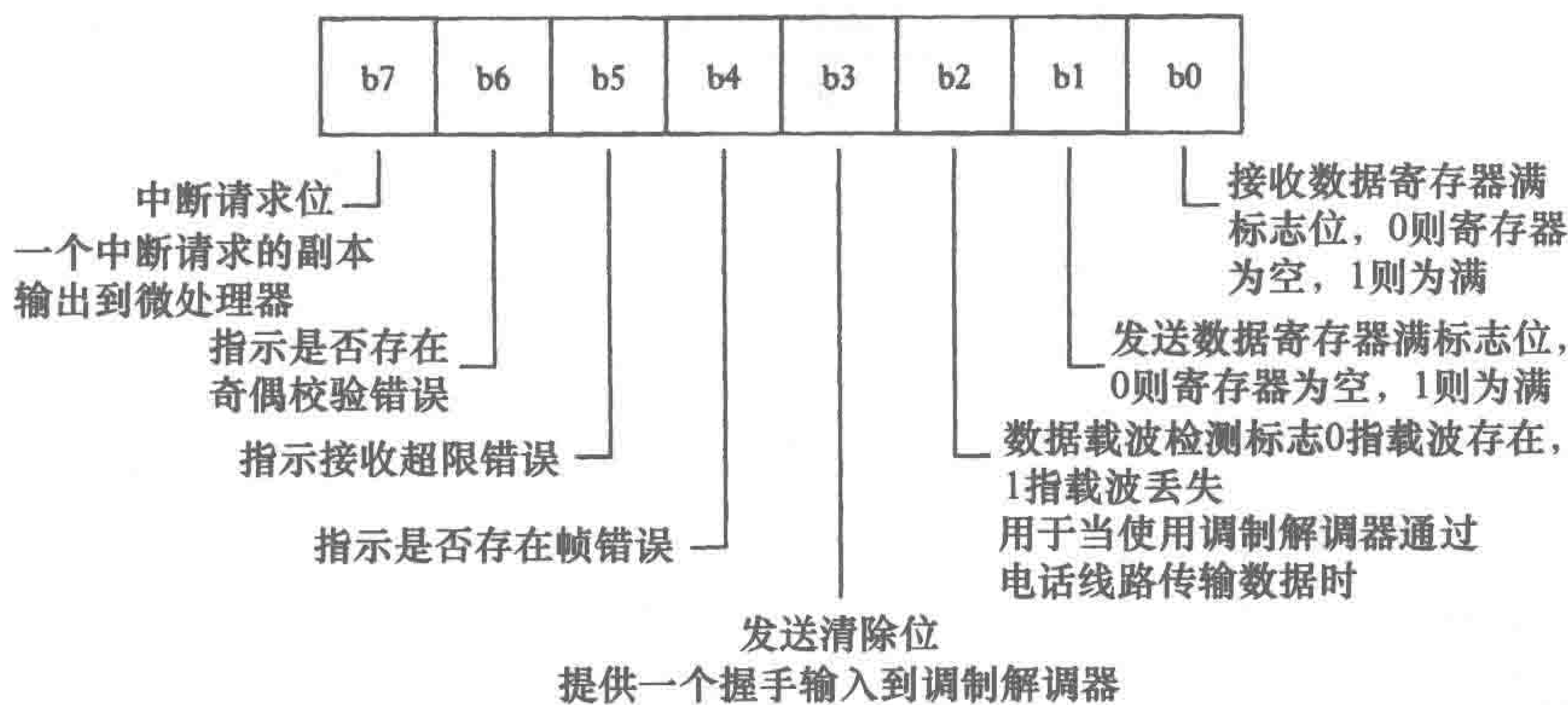


图 13-16 状态寄存器

13.5.1 微处理器的串行接口

许多微控制器都有串行接口，即内置的 UART。例如，摩托罗拉 M68HC11 有一个串行外设接口（SPI）、一个同步接口、一个串行通信接口（SCI）和一个异步接口（见图 10-10）。SPI 需要微控制器和外部连接的设备使用相同的时钟信号（见图 13-17a）。SPI 允许多个微控制器通过它相互联系起来。SCI 是异步接口，允许 SCI 系统和外部连接的设备使用不同的时钟信号（见图 13-17b）。通用微处理器没有 SCI 接口，如摩托罗拉 MC6850，使用 UART 替代来进行串行通信。在某些情况下，需要一个以上的 SCI 接口，因此微控制器 M68HC11 需要扩展 UART 接口。

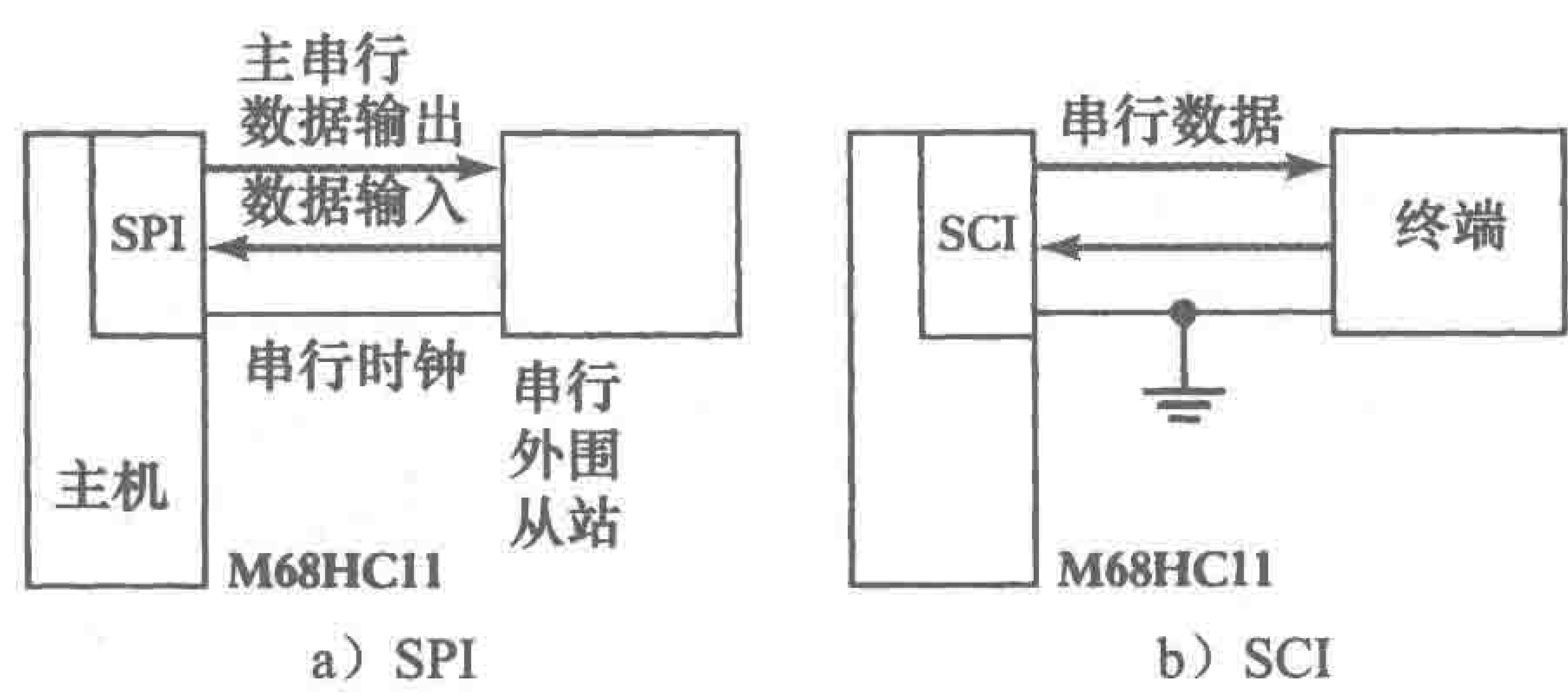


图 13-17 串行通信接口

SPI 接口由 SPI 控制寄存器（SPCR）和端口 D 的数据方向控制寄存器（DDRD）中的位初始化。SPI 状态寄存器包含状态位和错误位。SCI 接口由 SCI 控制寄存器 1、SCI 控制寄存器 2 以及波特率控制寄存器初始化。状态标志位于 SCI 状态寄存器中。

Intel 8051 有一个内置的串行接口，它有 4 种操作模式的，通过向位于地址 98H 的 SCON（串口控制）寄存器的 SM0 和 SM1 位写入 1 或 0 来选择（见图 13-18，表 13-1）。

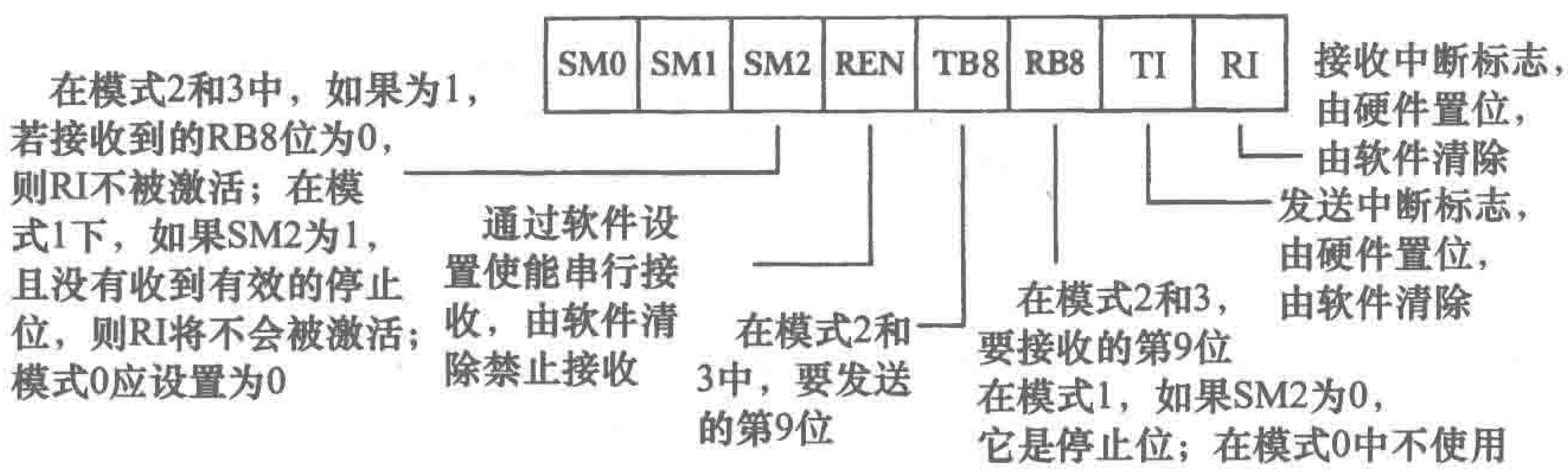


图 13-18 SCON 寄存器

表 13-1 英特尔 8051 串口模式

SM0	SM1	模式	描述	波特率
0	0	0	移位寄存器	晶振频率/12
0	1	1	8 位 UART	可变
1	0	2	9 位 UART	晶振频率/32 或 64
1	1	3	9 位 UART	可变

在模式 0 中，串行数据通过 RXD 进入和离开。TXD 引脚输出移位时钟，然后将它用于同步数据发送和接收。当 REN 置为 1 和 RI 置为 0 时，开始接收。任何数据写入到位于地址 99H 的串口缓冲器 SBUF 时，启动发送。在模式 1 中，10 位数据通过 TXD 发送或者通过 RXD 接收，这些数据都是起始位为 0，8 个数据位，停止位为 1。写入 SBUF，开始发送，由 RXD 上一个下降沿开始接收。在模式 2 和 3 中，11 位的数据通过 TXD 发送，RXD 接收。

PIC 微控制器有一个可用于同步串行通信的 SPI 接口（见图 10-30）。当数据写入 SSBUF 寄存器时，它与 SCK 引脚上的时钟信号同步移出 SDO 引脚，并作为串行信号，通过引脚 RC5 高位在前输出，时钟信号通过引脚 RC3 输出。通过引脚 RC4 输入到 SSBUF 寄存器。许多 PIC 微控制器也有一个 UART 串行接口，用来异步串行数据传输。发送时，每 8 位数据加一个起始位和一个停止位构成一帧。起始位传输时，RX 线降为低电平，接收器可以通过这个下降沿同步。然后接收器读取 8 位串行数据。

13.6 接口的示例

以下是接口的一些例子。

13.6.1 通过译码器控制七段数码管

343~344

考虑从微处理器的输出来驱动一个 LED 显示单元（见 6.5 节）。一个 LED 单元是一个开/关指示器，LED 单元的开决定了显示的数字。图 13-19 显示了如何运用微处理器和译码器来驱动共阳极数码管，只需输入 BCD 码，译码器就会自动转换成用于显示的对应的控制码。

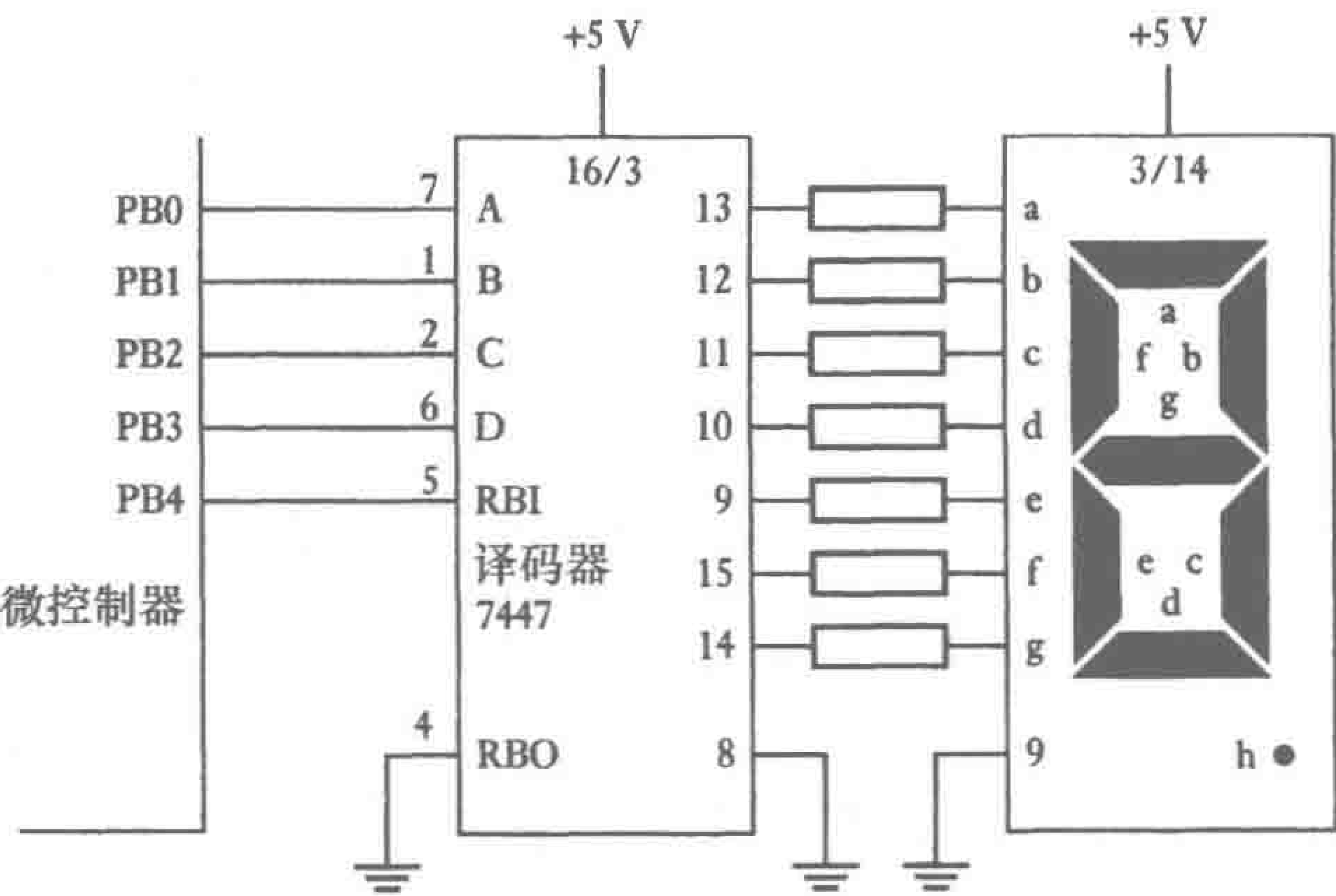


图 13-19 驱动显示器

对于 7447 译码器，引脚 7、1、2、6 是译码器的输入引脚，采用 BCD 码输入，引脚 13、12、11、10、9、15、14 作为驱动数码管的输出引脚。数码管的引脚 9 表示小数点，表 13-2 显示译码器的输入/输出信号。

表 13-2 七段显示 7447BCD 译码器

显示	输入引脚				输出引脚						
	6	2	1	7	13	12	11	10	9	15	14
0	L	L	L	L	ON	ON	ON	ON	ON	ON	OFF
1	L	L	L	H	OFF	ON	ON	OFF	OFF	OFF	OFF
2	L	L	H	L	ON	ON	OFF	ON	ON	OFF	ON
3	L	L	H	H	ON	ON	ON	ON	OFF	OFF	ON
4	L	H	L	L	OFF	ON	ON	OFF	OFF	ON	ON
5	L	H	H	L	ON	OFF	ON	ON	OFF	ON	ON
6	L	H	H	L	OFF	OFF	ON	ON	ON	ON	ON
7	L	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF
8	H	L	H	H	ON	ON	ON	OFF	OFF	OFF	OFF
9	H	L	H	L	ON	ON	ON	OFF	OFF	OFF	OFF

当数码管的每段都不亮时，称为数码管**空白**（或消隐）。当我们有 3 个数码管单元，希望显示 10 而不是 010，为了使得第一个数码管单元 0 不显示，我们需要使第一个数码管空白。通过使 RBI 引脚为低电平，数码管空白就能实现。当 RBI 为低电平和 BCD 输入 A、B、C、D 全为 0 时，输出是空白。不管 RBI 是什么状态，当输入不是 0 时，RBO 引脚总是输出高电平。第一位数码管的 RBO 引脚能够连接到第二位数码管的 RBI 引脚，第二位数码管的 RBO 引脚能够连接到第三位数码管的 RBI 引脚，这样能使得最后 RBO 引脚为 0 的前面数码管输出空白（见图 13-20）。

345

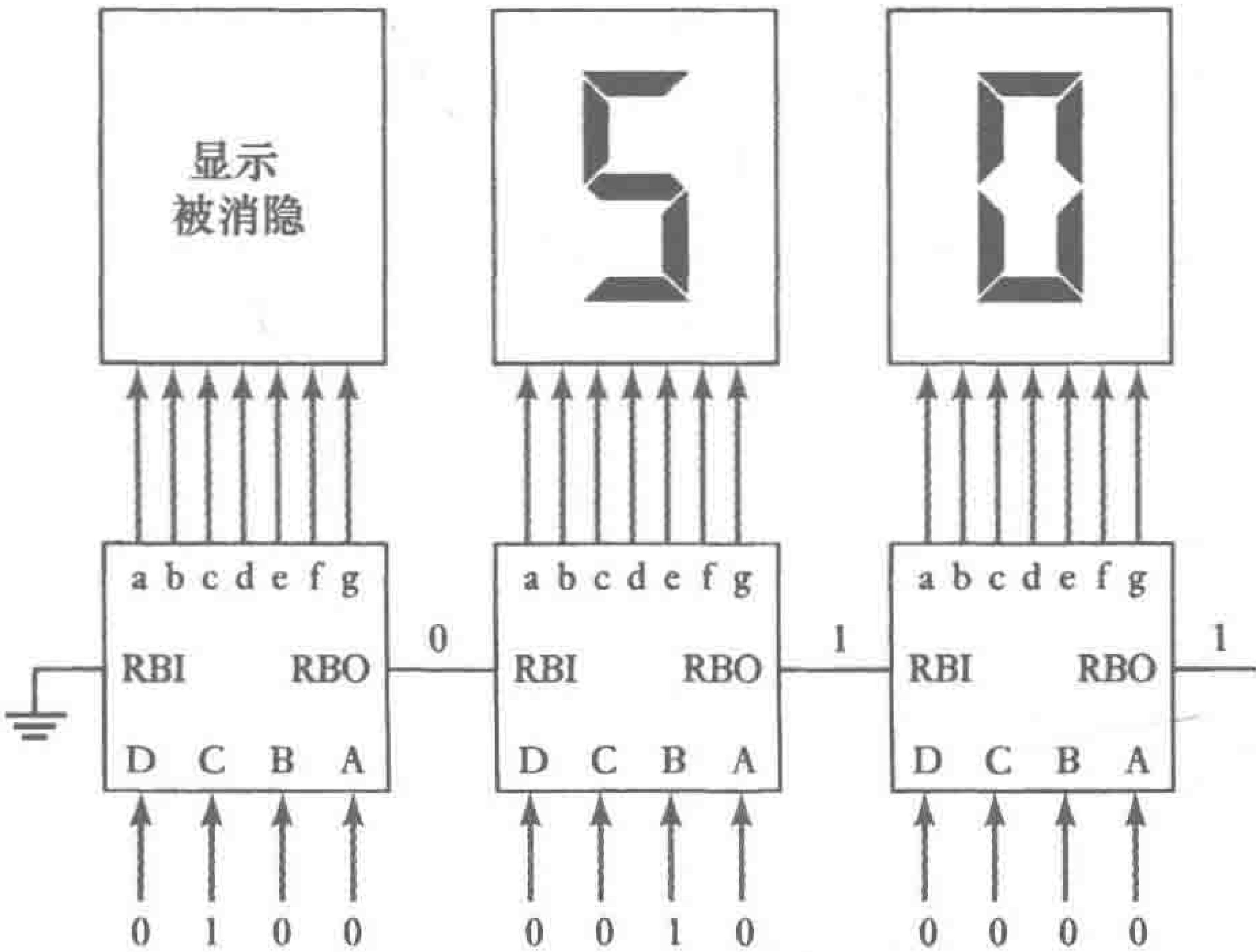


图 13-20 消隐

当需要显示很多数码管单元时，不需要为每一个单元配一个译码器，可以用多路开关通过单个译码器来控制多个数码管。图 13-21 显示了多路开关控制的 4 个共阴极数码管电路。控制器 A 部分输出 BCD 码，译码器译码后输出到显示，每个数码管通过晶体管作为共阴极连接到地。当端口 B 输出使得晶体管导通，对应的数码管才点亮显示。因此，通过切换 PB0、PB1、PB2、PB3 和端口 A 引脚能够获得合适的显示。为了获得一个稳定的显示，数码管快速的切换来实现视觉无闪烁的显示。时分复用可以用来一次显示多个数字。

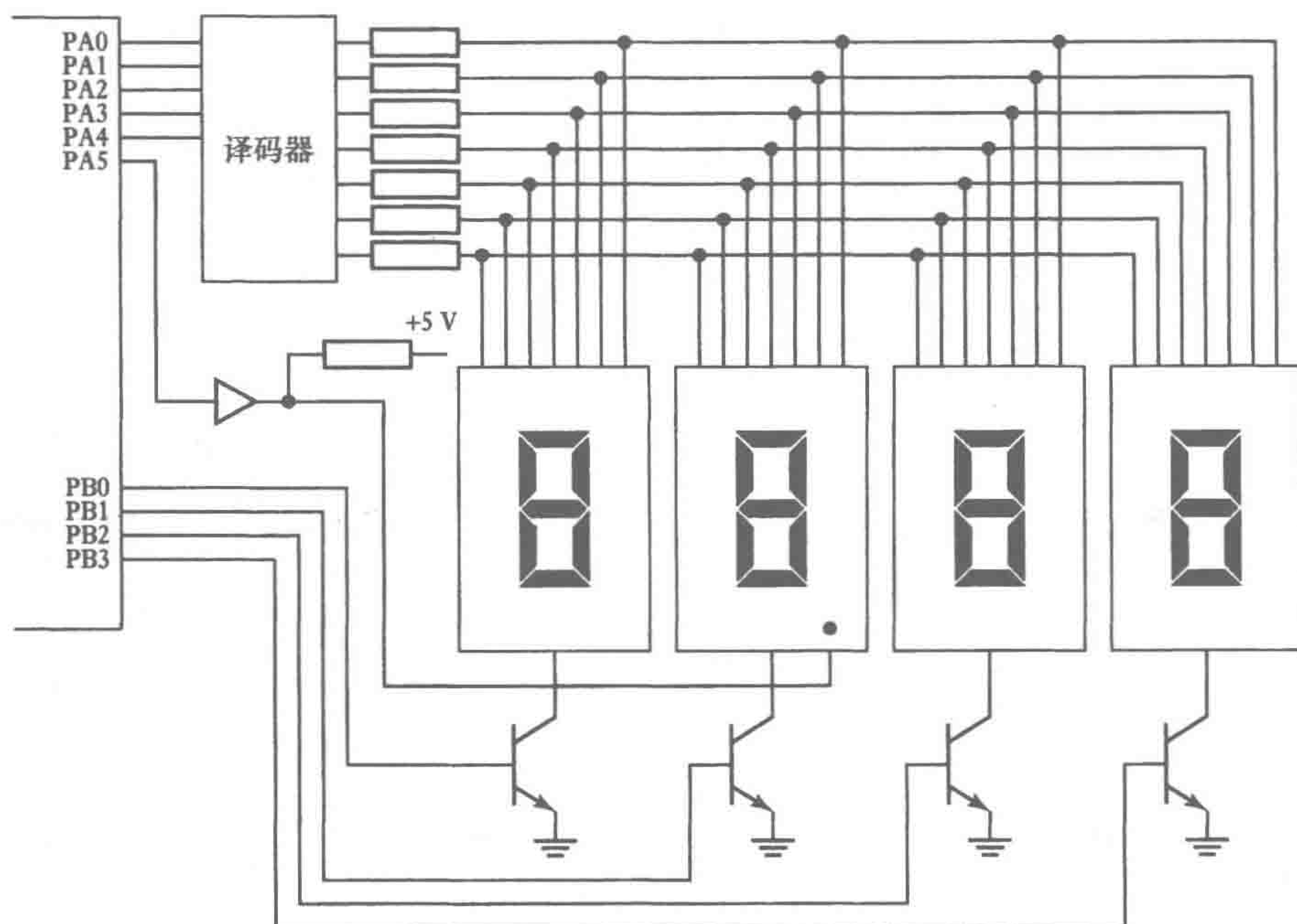


图 13-21　多路复用 4 个数码管

13.6.2　模拟接口

当微处理器或者微控制器需要提供模拟信号输出时，需要模数转换。比如，DAC 模拟设备 AD557 可以用于该用途。它能输出一个正比于数字输入的信号，同时输入锁存器可用于微处理器的接口。如果不需要锁存，引脚 9 和 10 就接地。当有一个上升沿时，数据会被锁存，即电平在引脚 9 或 10 从低到高变化。数据会被锁存直到这两个引脚电平重新变成低电平。当发生这种变化时，为了转换成模拟电压数据会从锁存器转移到 DAC。

图 13-22 显示 AD557 不使用锁存，与摩托罗拉 M68HC11 相连接，当下面的程序运行时，系统会产生一个锯齿形电压输出。其他的电压波形可以在此基础上修改程序来产生。

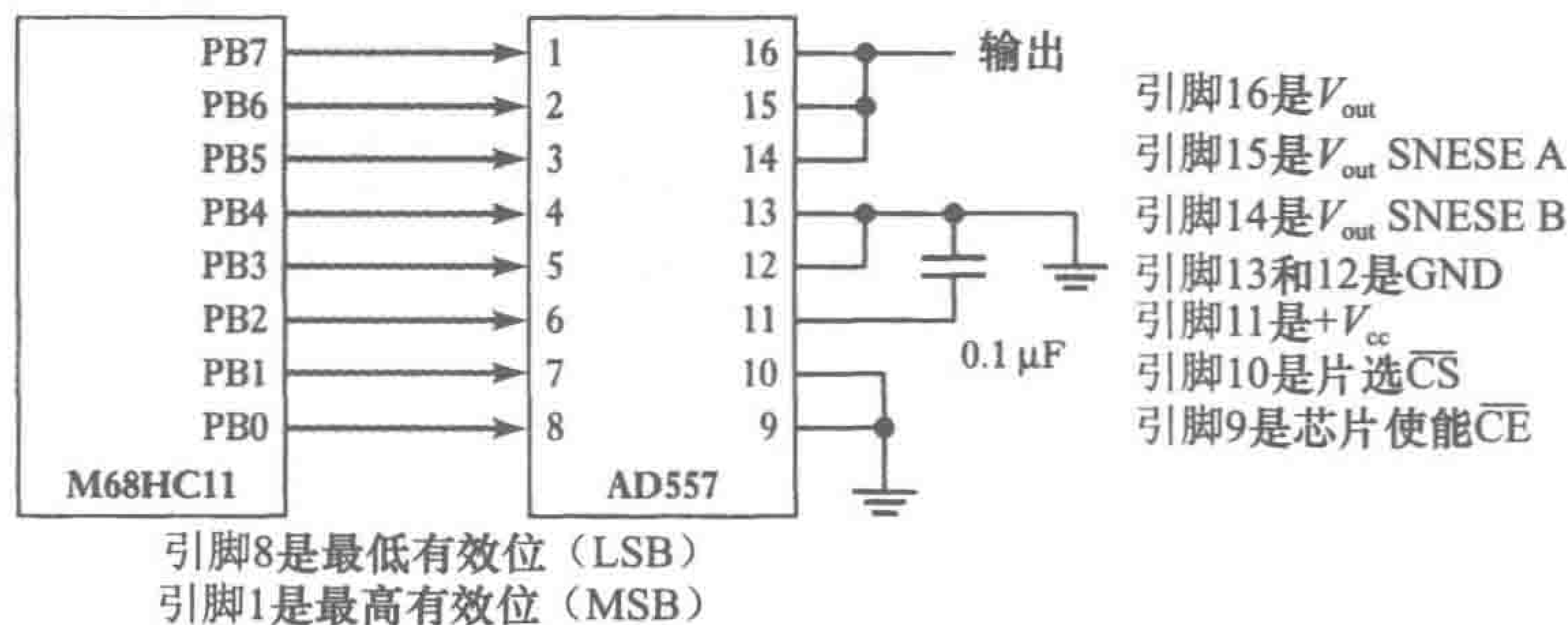


图 13-22　波形生成

```
BASE    EQU $1000      ;      I/O 寄存器的基地址
PORTB   EQU $04        ;      PORTB 相对于 BASE 的偏移量
        ORG $C000
        LDX #BASE      ;      X 指向寄存器地址
        CLR PORTB,X    ;      DAC 清零
AGAIN   INC PORTB,X    ;      加 1
        BRA AGAIN      ;      重复
        END
```

本章小结

接口要求常指电气缓冲/隔离、时序控制、代码转换、改变线数、串行变成并行、并行变成串行，模数转换和数模转换。除非两设备接收和发送数据的速率是相同的，否则**握手**是必需的。

346~347

轮询是指 I/O 的程序控制，应用循环连续读取输入更新输出，同时跳转到所需的服务程序，即重复检查外部设备看它们是否做好准备接收或者发送新数据的过程。另一个程序控制是**中断控制**。中断包括外部设备触发一个独立的中断请求线。微处理器接收到中断请求后，暂停执行主程序，跳转到外设服务程序。中断服务程序结束后，内存中保护的数据恢复，微处理器可以继续在原中断处继续执行主程序。

有两种基本的串行数据传输方式：异步和同步。当**异步传输**时，接收器和发送器各自使用自身的时钟信号，所以接收器是不可能知道何时数据发送开始或者已经结束了。因此有必要使每个发送帧带有自己的起始位和结束位，从而可以让接收器辨别一个数据帧的结束和另一个数据帧的开始。对于**同步传输**，接收器和发送器有共同的时钟信号，所以发送和接收可实现同步。

外部接口适配器（PIA）是可编程的输入/输出接口设备，它可以通过软件选择各种不同的输入和输出选项。

通用异步接收/发送器（UART）是串行通信系统中基本的元素，它的功能是串行数据变为并行输入和并行数据变为串行输出。一种常见的 UART 的可编程形式是异步通信接口适配器（ACIA）。

习题

13.1　描述一个接口所需要的功能。

13.2　解释串行和并行接口的区别。

13.3　解释什么是用于输入和输出的内存映射系统。

13.4　描述外围接口适配器的功能。

13.5　描述摩托罗拉 MC6821 PIA 的结构。

13.6　解释外部接口适配器初始化程序的功能。

13.7　作为与外部设备的交互，外部中断比程序轮询有什么优势？

13.8　对于摩托罗拉 MC6821 的外部接口适配器，如果 CA1 不使能，则 CB1 有一个上升沿设置的使能中断输入，CA2 使能且用来作为设置/重置输出，CB2 使能且在第一个上升沿变为低电平，随后微处理器写数据到外部数据存储器 B，并在下一个上升沿变成高电平，什么值应该存储在控制存储器中？

13.9　用汇编语言写摩托罗拉 MC6821PIA 的初始化程序，实现习题 13.8 所给出的功能。

13.10　用汇编语言写摩托罗拉 MC6821PIA 的初始化程序，实现对端口 A 读 8 位数据。

348

第14章　可编程逻辑控制器

本章目的

通过本章的学习，读者将能够：

- 描述PLC的基本结构以及它们的操作。
- 能够设计包含基本逻辑功能、自锁、内部辅助继电器、时序的梯形图。
- 能够设计包含定时器、计数器、移位寄存器、主控继电器、跳转以及数据处理的程序。

14.1　简介

可编程逻辑控制器（PLC）是一个能够使用可编程内存来存储并执行逻辑、顺序控制、定时、计数、算术等功能的电子设备，它能够控制机器以及相应的过程，实现简单、高效的编程。术语逻辑的使用是因为编程主要是涉及逻辑与开关操作。输入部分，例如开关，输出部分，例如电动机，被连接到PLC，然后控制器根据操作者存储在PLC中的程序监控输入和输出，从而控制机器和过程。最初的PLC是用来取代硬线连接继电器（见图9-2）和定时器逻辑的控制系统。PLC可以在不重新改动输入/输出设备硬件接线的情况下实现控制系统的修改，唯一的要求就是操作者输入一些基本的指令，因此PLC具备很大的优势。另外，与继电器操作的系统相比，PLC响应非常快。PLC的应用使复杂而灵活多变的控制系统成为可能，所以目前应用非常广泛。

PLC和计算机类似，但是它们作为专用控制器又具有一定的特征。它们是：

1）更加耐用，并且被设计得具有抗震动，适应一定的温度、湿度、噪声等比较恶劣的环境。

2）输入和输出接口都在控制器内部。

3）编程简单。

14.2　PLC基本结构

图14-1表示PLC的基本内部结构。它包含一个中央处理器（CPU）、内存以及输入/输出接口。CPU在内部控制并处理所有的操作。通常它具有一个频率在1～8MHz的时钟。这个频率决定了PLC的工作速度并且为所有的系统单元提供定时以及同步功能。总线系统在CPU、内存、输入/输出单元之间传递数据。系统的内存单元有：系统的程序存储器ROM存储永久的数据和程序指令，数据存储器RAM存储用户的程序，并且为输入/输出通道提供临时缓冲区。

349

14.2.1　输入/输出

输入/输出单元提供了系统内部和外部的接口，处理器能够从外部设备接收信息，并能够向外部设备发送信息。输入/输出接口提供了隔离和信号处理功能，这样传感器以及驱动器可以直接连接到输入/输出上，而不用另外的电路。输入部分可以是由外部事件触发的限位开关，也可以是温度、流量等传感器。输出部分可能是电动机启动线圈或者电磁阀等。与外界的电气隔离通常用光耦来实现（参见3.3节）。

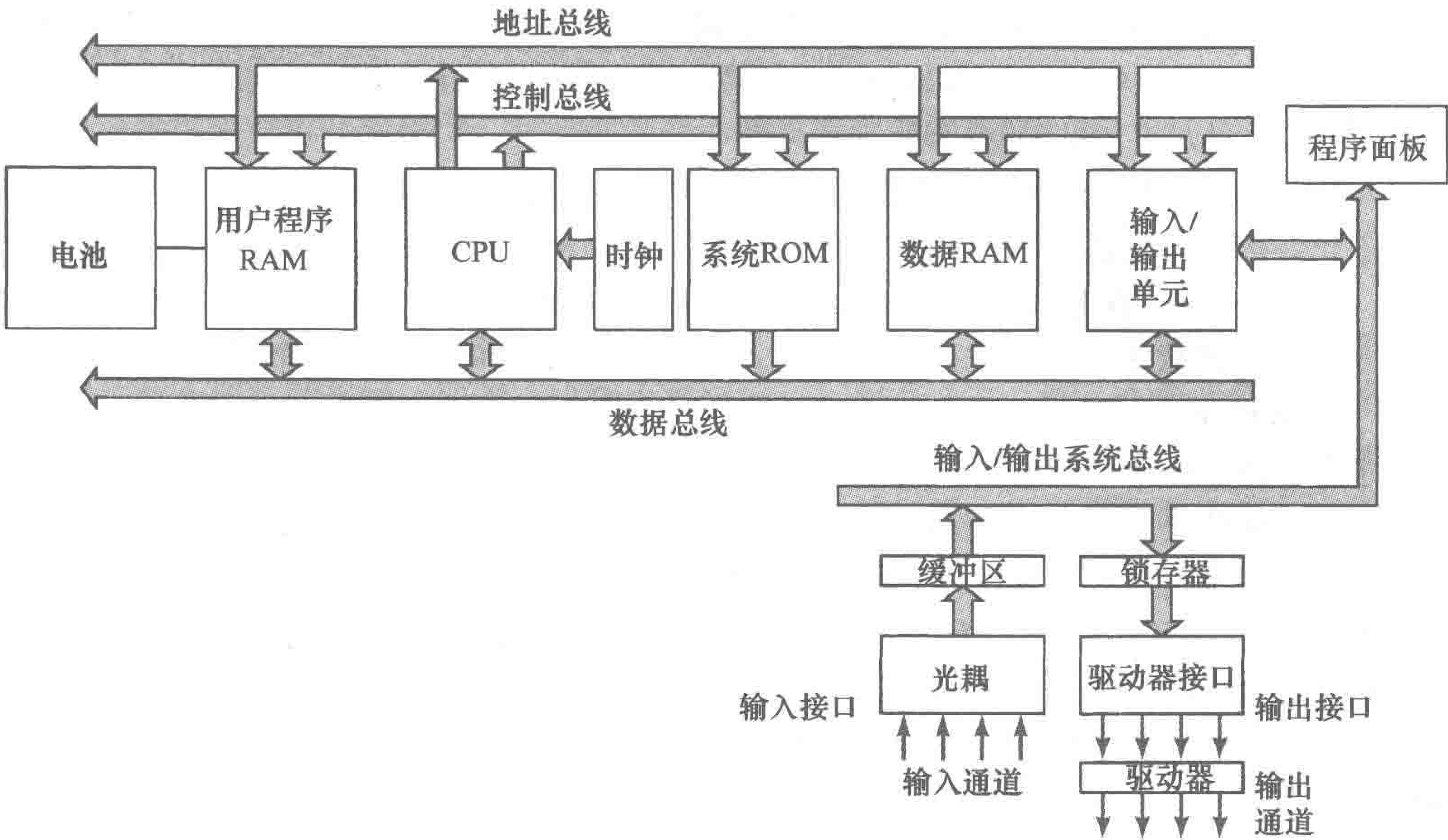


图 14-1 可编程逻辑控制器的结构

图 14-2 说明了输入通道的基本形式。与 PLC 中的微控制器兼容的数字信号通常为直流 5V。然而，通过在内部通道的信号调理和隔离，能够提供很大范围内的输入信号。这样，较大的 PLC 可有 5V、24V、110V 以及 240V 的输入电压。一个较小的 PLC 则通常只有一种形式的输入，例如 24V。

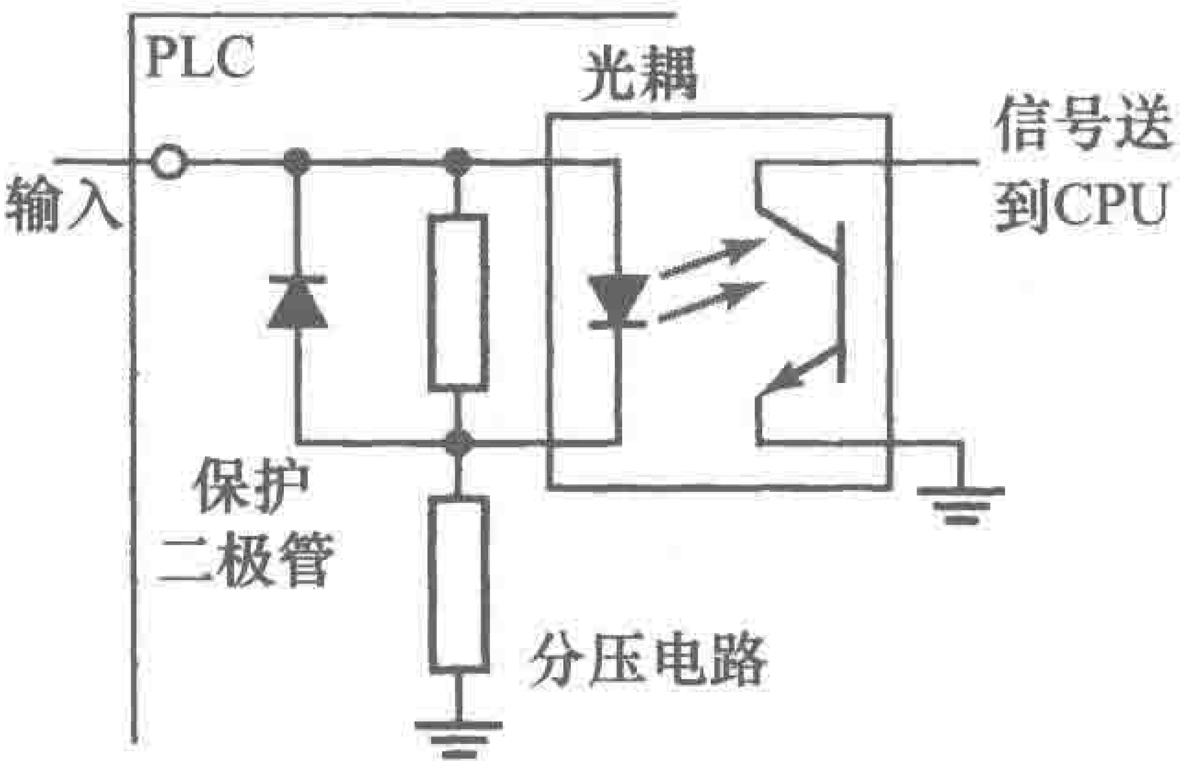


图 14-2 输入通道

输出到输出单元的数字信号一般为 5V 电平。输出类型被分为继电器输出、晶体管输出以及晶闸管输出。对于继电器输出，PLC 的输出信号主要在内部控制一个继电器，这样能够控制外部电路中几个安培的电流。继电器把 PLC 与外部电路相隔离，可以用于交流与直流电路的切换。不过，继电器输出的速度比较慢。晶体管的输出方式使用晶体管控制外部电路电流通断，给出更快的切换动作。光耦合器用于外部电路与 PLC 的电气隔离。晶体管输出只是用于直流切换。晶闸管输出可以控制与交流电源连接的外部负载。我们可通过 24V、100mA 的开关信号控制一个直流 110V、1A，或一个交流 240V、1A，或交流 240V、2A 的来自晶闸管输出通道的负载。对于较小的 PLC，通常只有 240V 的交流输出。而一个模块化的 PLC 可以通过选择不同的模块适应很大范围的输出信号。

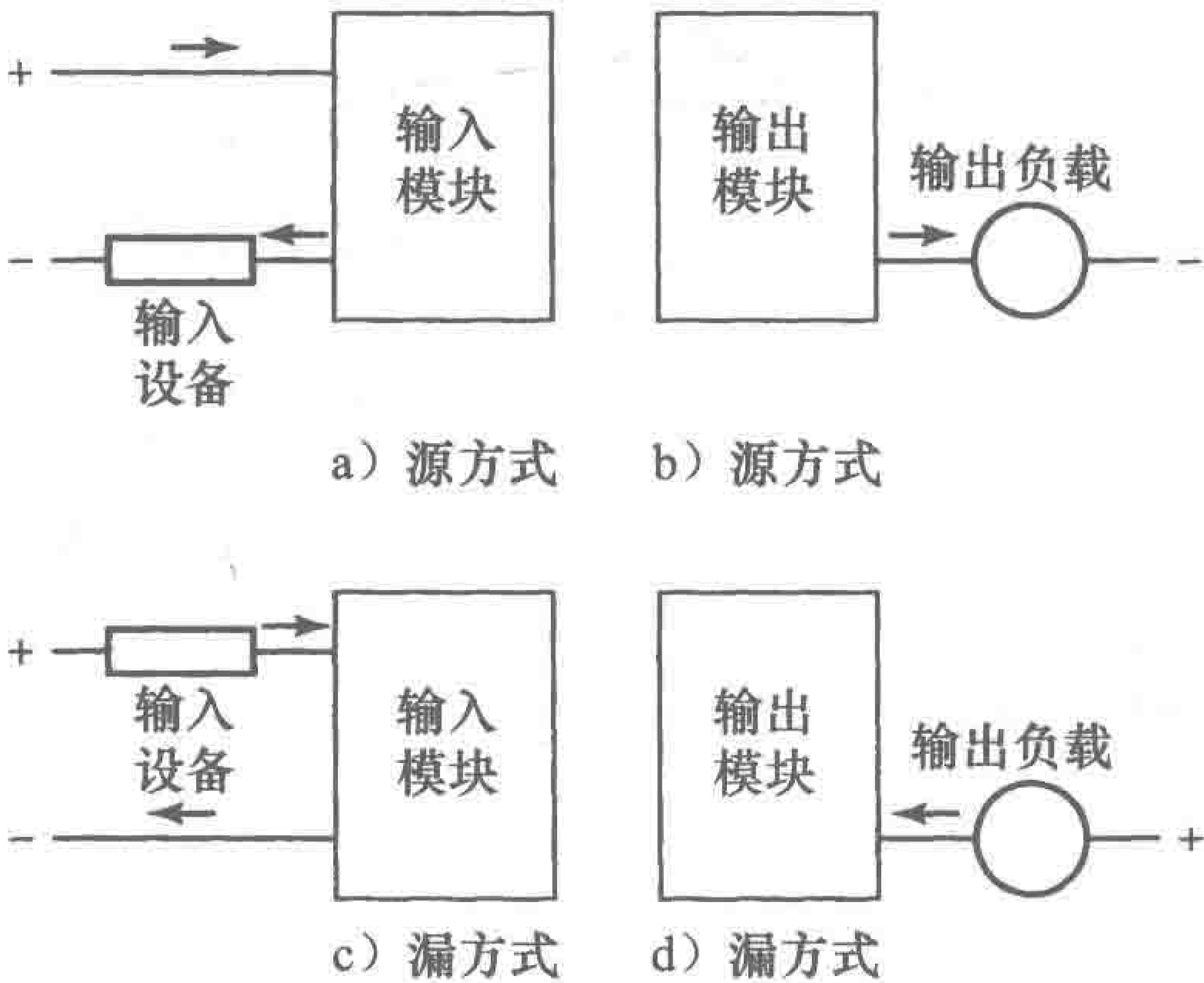


图 14-3 连接方式

源（sourcing）和**漏**（sinking）描述了直流设备与 PLC 的连接方式。通过“源方式”，用传统的电流流动方向（正极到负极），输入设备从输入模块中得到电流（见图 14-3a）。如果电流流向是输

出模块向输出负载，那么输出模块也认为是处于源方式（见图 14-3b）。对于“漏方式”，一个输入设备向输入模块提供电流（见图 14-3c）。如果电流的流向是输出负载向输出模块，那么输出模块也认为是处于漏方式（见图 14-3d）。

输入/输出单元提供了系统与外部世界的接口，输入通道允许接输入设备（如传感器），输出通道允许接输出设备（如电动机和线圈）。通过输入/输出单元，程序也可以通过程序面板输入。每一个输入/输出点都有一个唯一的被 CPU 使用的地址。就像一条马路边上的一排房子：10 号房子可能代表一个特殊传感器的输入，而 45 号房子可能代表一个特定电动机的输出。

14.2.2　输入程序

程序通常通过一个小的便携式的编程器（编程器带有可视化显示单元（VDU）的桌面控制台、键盘和屏幕显示器），或者通过连接安装特定软件的个人计算机（PC）输入到输入/输出单元。只有在编程器上程序设计完成后才能传输到 PLC 的内存中。

RAM 中的程序可以被用户修改。但是，为了防止断电时程序丢失，电池会给 PLC 供电一段时间来保持 RAM 内容。当程序在 RAM 中开发好后，它有可能载入到 EPROM 芯片中并得到永久保存。PLC 指标通常指定了可以存储程序步的数量的程序存储器的大小。程序步是发生某个事件的指令。一个程序任务可能含有多个程序步，例如可能首先检查开关 A 的状态，再检查开关 B 的状态，如果 A 和 B 都处于闭合状态则给线圈 P 上电，使某些驱动器动作。通常一个小 PLC 可以执行的程序步的数量级是 300 ~ 1000，而这个范围也能够满足绝大多数的控制需求。

14.2.3　PLC 的形式

PLC 于 1968 年首先被提出。现在已获得了广泛应用，并且从小的独立单元，例如能够使用 20 个数字输入/输出的单箱结构，发展到带有大量输入/输出，并且能够处理模拟或数字输入/输出以及比例积分微分（PID）控制的机架安装系统。单箱型通常用于较小的可编程控制器，配备有紧凑但是完整的电源输入、处理器、内存、输入/输出单元。这样的 PLC 通常含有 6、8、12 或 24 个输入以及 4、8 或 16 个输出，并且能够存储 300 ~ 1000 条程序指令。例如 MELSEC FX3U 的模块有 6、8、12 或 24 个输入和 4、8 或 16 个继电器输出，能存储 300 ~ 1000 条指令。有些系统可以通过连接 I/O 扩展模块来处理更多的输入/输出。

具有较多输入/输出的系统通常都是模块化的。包含有独立的电源、处理器、输入/输出等模块，这些模块在金属箱里通过导轨连接起来。机架的类型可以用于所有大小的可编程控制器，有封装成独立模块的可插入机架插槽的各种功能单元。各种功能模块的选用通常由用户来决定，然后插入到机架上即可。所以，增加输入/输出可以通过添加输入/输出模块简单地实现。例如，SIMATIC S7-300/400 PLC 就是这种机架安装 PLC，包括电源、CPU、I/O 接口模块、I/O 信号调理模块、PLC 与其他系统的通信模块等组件。

350～352

另一个模块化的系统的例子是 Allen-Bradley SLC-500 可编程逻辑控制系统。这是一个体积小、框架式、有若干电源供给模块，以及大量 I/O 端口和多处理器选择的可编程控制器模块化系列。SLC-500 允许针对一个应用专门建立一个系统。PLC 模块安装在机架上，不同的模块通过背板的总线互相连接。PLC 的电源是机架上末端的盒子，而另外一个盒子包含一个微处理器。背板的总线含有铜导线，这样嵌入机架槽中的模块能够接入电源，并且模块和处理器之间能够相互交换数据。模块滑入机架并且卡住背板上的连接器。SLC-500 系列 PLC 的机架能够安装 4、7、10 或者 13 个模块。这些模块能够提供 8、16、32 个电流“漏”（sinking）方式直流输入，8、16 或 32 个电流“源”方式直流输出，4、8 或 16 个继电器直流/交流输出，与计算机或者其他 PLC 通信的通信模块。通常情况下，软件都可以在 Windows 平台上编程。

14.3 输入/输出处理

PLC 是连续运行内部的程序，并且根据输入信号的结果来刷新。每一个这样的循环称为**周期**。通常有两种输入/输出的处理方法：持续更新（continuous updating）和大规模输入/输出复制（mass input/output copying）。

14.3.1 持续更新

持续更新包含 CPU 扫描程序指令中出现的输入通道。每一个输入点都被单独检查，并且决定它对于整个程序的影响。通常会出现内部延时，一般为 3ms，每一个输入都会被检查以保证可靠的输入数据能够被微处理器使用。如果开关在接触时出现抖动，那么延时使控制器能够避免把一个输入计数两次或者更多次。多个输入或许会被扫描，在有逻辑操作指令的程序执行或者产生外部输出之前，每一个输入扫描都会有 3ms 的延迟。输出被锁存以保持再次刷新之前它们已有的状态。

14.3.2 大规模输入/输出复制

由于持续更新的时候每一个输入点都有 3ms 的延迟，所以检验数百个输入/输出点的时间将非常长。为了实现快速的程序执行，将一块特定的 RAM 用做控制逻辑以及输入/输出之间的缓冲区。每一路输入/输出在这个内存中都有相应的地址。在每个程序周期开始时 CPU 扫描所有的输入/输出状态扫描并将它们复制到 RAM 中的对应地址。当程序执行时，要求存储的输入数据按要求从 RAM 以及所执行的逻辑运算中读取。结果存储在 RAM 中保留的输入/输出部分。 353 在每一个程序循环的末尾，所有的输出都会从 RAM 转移到输出通道。输出会被锁存，这样它们可以在下一次刷新之前保持原有的状态。具体顺序如下：

1）扫描所有的输入，复制到 RAM 中。

2）顺序取指、解码并执行所有的程序指令，并把所有的输出复制到 RAM 中。

3）刷新所有的输出。

4）重复上述过程。

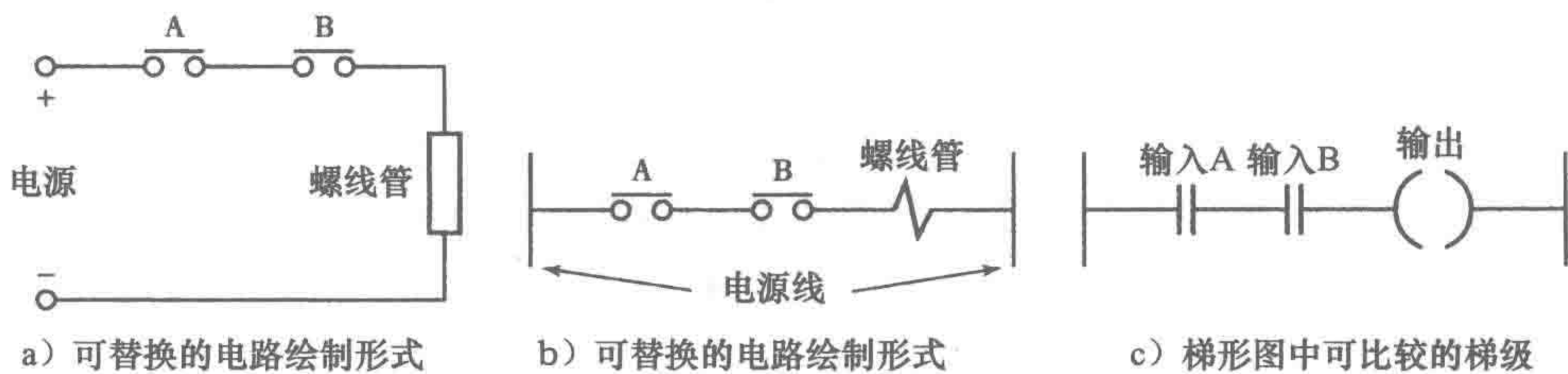
PLC 根据程序指令花费一定的时间来完成一个周期的输入扫描和输出刷新，因此输入并没有被时时刻刻地监视而只有定时地检查。典型的 PLC 周期通常需要 10～50ms，因此输入和输出也是每 10～50ms 刷新一次。这意味着如果一个非常简短的输入出现在周期的错误时刻，那么它可能不会被读取。因此，对于一个周期时间为 40ms 的 PLC，最大的可以探测到的数字周期为每 40ms 出现的脉冲。对于 Mitsubishi 紧凑型 PLC，MELSEC FX3U，每个基本逻辑指令有 0.065μs 的程序周期时间，因此如果程序越复杂，那么程序周期也就越长。

14.3.3 输入/输出地址

PLC 必须能够识别每一位的输入与输出（I/O），这是通过给每一个 I/O 分配地址来实现的，与同一个镇上的房子一样，只有通过它独有的地址才能把东西邮寄到正确的家庭。例如，Mitsubishi 和 Toshiba 的输入一般表示为 X400、X401、X402 等，输出表示为 Y430、Y431 等。对于较大的 PLC，如果有多个机架的输入/输出通道，每个机架上有多个输入/输出模块，则对机架和模块进行编号，因此输入/输出通过机架号和后面的模块号来识别，然后用一个数字表示模块的端子号。例如，Allen-Bradley PCL-5 使用 I:012/03 表示一个输入在 01 机架、模块 2 上，并且端子号为 03。

14.4 梯形图编程

PLC 中常用的编程形式是**梯形图编程**。每一个程序任务就像梯子中的每一级横档。这样的横档可以确定开关 A、B 的状态，如果 A、B 均为闭合，那么输出的电磁阀就会上电。图 14-4 就是电路实现方式与梯形图的比较。



354

图 14-4　电路实现方式与梯形图的比较

PLC 执行程序的顺序可以概括为：

1）扫描梯形图中每一梯级的输入。

2）求解含有这些输入的逻辑运算。

3）置位/复位这些梯级的输出。

4）进入下一个梯级，重复执行操作 1、2、3。

5）进入下一个梯级，重复执行操作 1、2、3。

6）进入下一个梯级，重复执行操作 1、2、3。

7）程序一直执行到梯形图的最后一行，然后 PLC 回到程序的最开始重新执行。

基于梯形图的 PLC 编程非常类似于画开关电路的方式。梯形图的两条垂直方向的线代表了电源线。电路连接成水平方向的横线，就像梯子的横档一样，位于两条竖线之间。图 14-5 给出了基本的标准符号和梯形图梯级的例子。在绘制梯形图的梯级过程中，输入必须在输出之前，并且每行至少有一个输出。每个梯级必须起始于一个输入或者一系列输入，结尾必须是一个输出。

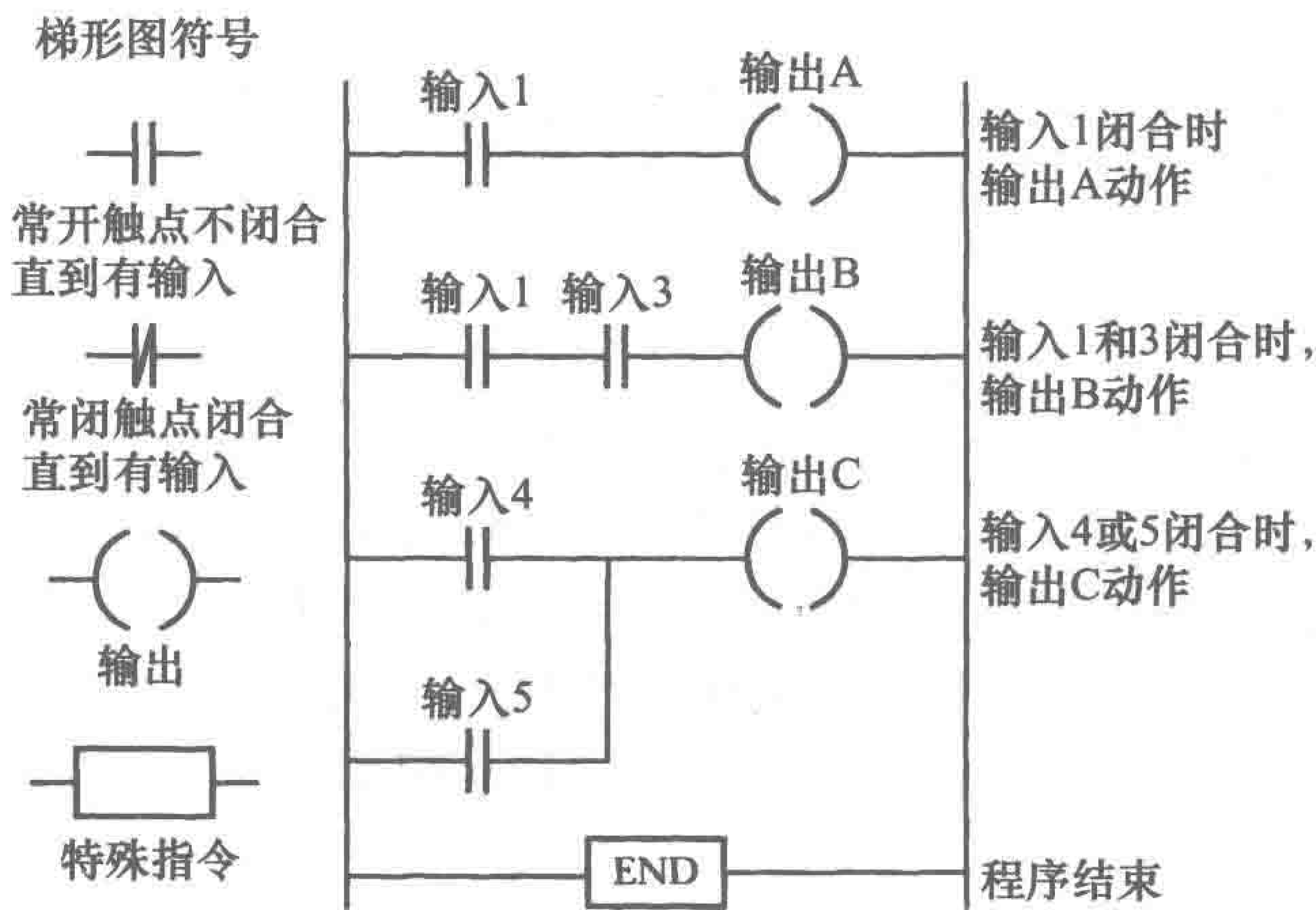


图 14-5　梯形图程序

为了说明如何画梯形图，考虑一种情况，其中 PLC 的输出端在一个常开触点闭合的情况下给线圈上电（见图 14-6a），所需要的梯形图程序如图 14-6b 所示。起始于一个输入，也就是常开符号 | | 。输入的地址为 X400。横线的结尾是输出，即电磁线圈，地址为 Y430。为了说明程序已经结束，用 END 来表示中止。当开关闭合时螺线管得电。例如，这可能用来打开电磁阀允许水进入容器内。

355

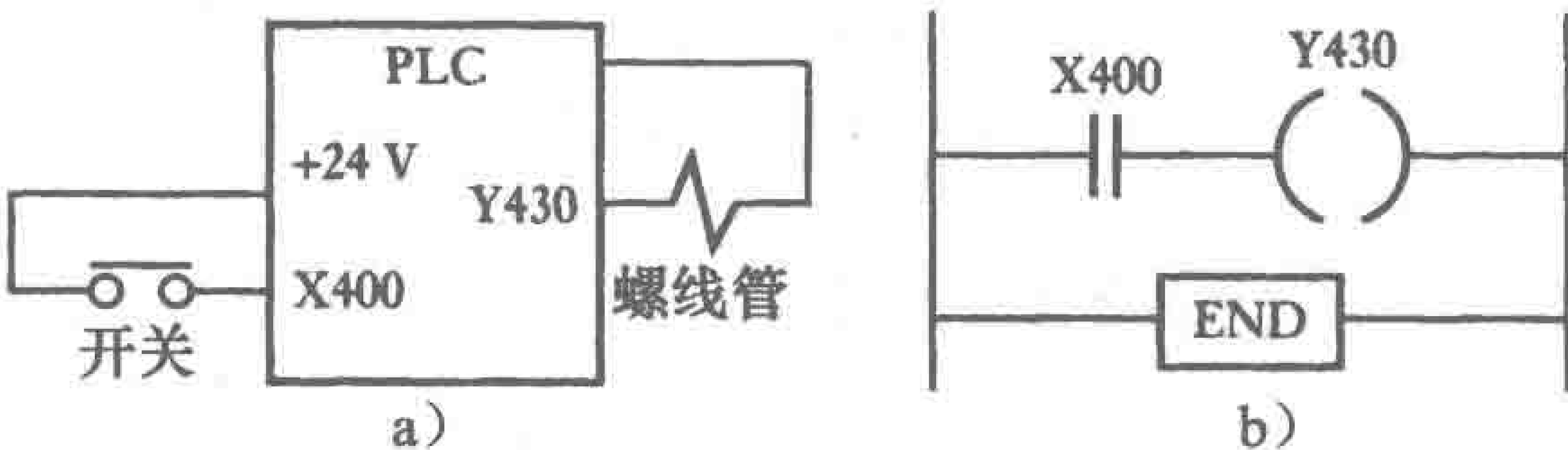


图 14-6　开关控制螺线管

另一个例子是开/关的温度控制（见图 14-7a）。当温度传感器达到设定的温度时，输入会从低到高变化。然后输出会从开变到关。

图 14-7中的温度传感器是一个连接在桥式电路上的热敏电阻，输出连接到作为比较器的运算放大器上（参见 3.2.7 节）。程序（见图 14-7b）中的输入是一个常闭触点，可以直接给出开信号和一个输出。当触点打开时给出关信号，关闭输出。

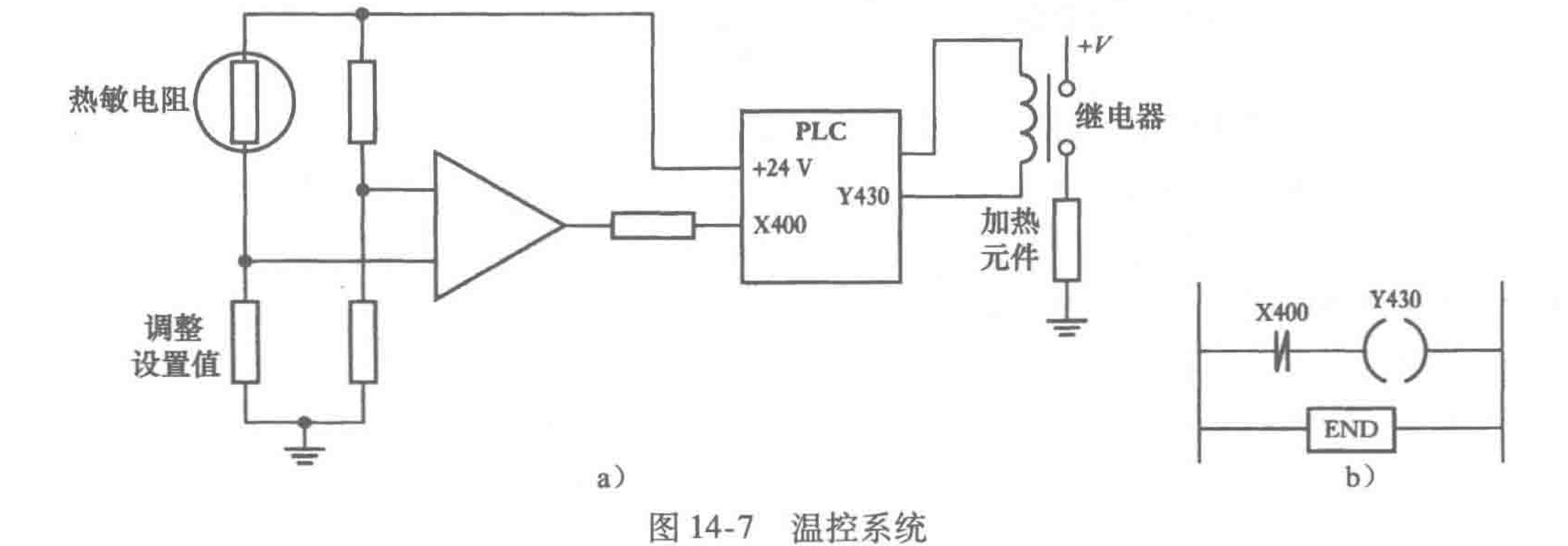


图 14-7 温控系统

14.4.1 逻辑函数

逻辑函数可以通过开关的组合来获得（参见 5.2 节），后面的部分会讲解如何用梯形图写这样的组合。

1. 与（AND）

图 14-8a 表示了当且仅当两个常开开关同时闭合，线圈才能够接通的情形。开关 A 和 B 必须同时闭合，这样就给出了 AND 的逻辑形式。等效的梯形图以 || 开始（输入 1 代表开关 A），串联 ||（输入 2 代表开关 B）。该行以 () 表示输出的结束。

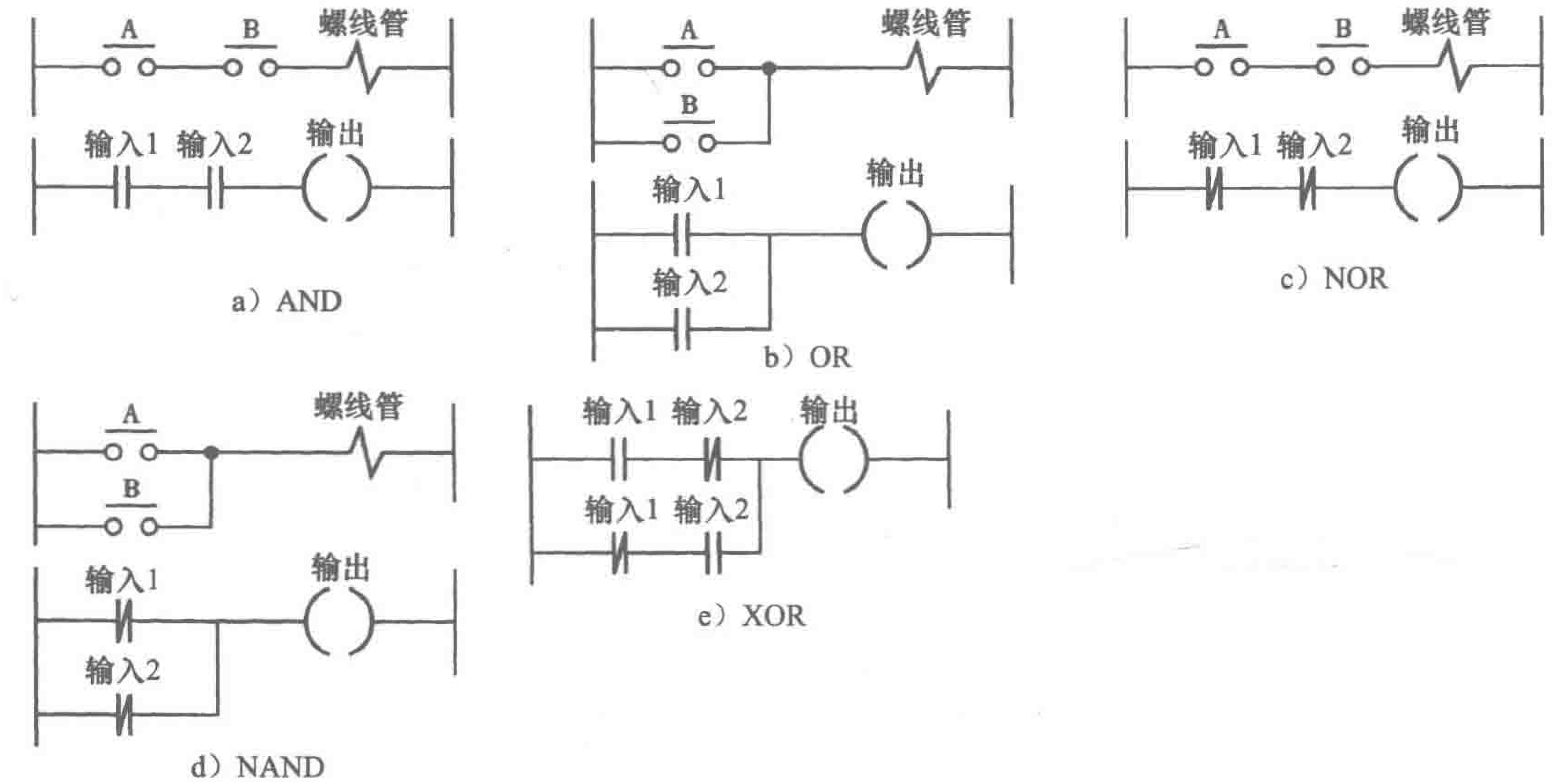


图 14-8 AND 逻辑

2. 或（OR）

图 14-8b 展示了当开关 A 或 B 中的一个闭合时线圈得电的情形。这就是 OR 逻辑门。等效的梯形图以 || 开始（输入 1 代表开关 A），并联 ||（输入 2 代表开关 B）。该行以 () 表示输出的结束。

3. 或非（NOR）

图 14-8c 表示了或非门的逻辑结构。开关 A 和 B 都没有输入时，才会有输出；当 A 或 B 中

的任意一个有输入时，输出停止。梯形图中输入 1 与输入 2 串联，都由常闭触点表示。

4. 与非（NAND）

图 14-8d 表示与非门。当且仅当开关 A 和 B 都有输入时，才没有输出。因此，对于梯形图程序，为了获得一个输出，要求输入 1 和输入 2 中至少有一个没有输入。

5. 异或（XOR）

图 14-8e 表示了如何用梯形图画异或门，当输入 1 和输入 2 都没有输入时，以及当输入 1 和输入 2 都有输入时，没有输出。注意，我们用常开和常闭两种触点表示每个输入。

考虑一种情况，如果一个常开开关 A 必须按下，而另外两个开关 B、C 中的任意一个必须按下时才能使线圈得电。我们可以画出电路（见图 14-9a），图中 B 与 C 并联，再与 A 串联。为了让线圈得电，必须让 A 闭合，或 B 与 C 中的一个闭合。考虑开关 A 和后面的两个开关串联给出了一个 AND 逻辑，B 和 C 并联给出了一个 OR 逻辑。于是我们有两种门的组合逻辑，其真值表如下：

输入			输出
A	B	C	
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

356～357

对于梯形图，我们以 | | 开始，标号输入 1 表示开关 A。然后与标号为输入 2 和输入 3（表示 B 和 C）并联的结果串联。这一行用（）结束来表示线圈输出。梯形图如图 14-9b 所示。

再考虑一个运用逻辑门的程序的简单例子。用一个螺线管控制阀门打开商店的门，当且仅当店主的控制开关打开，并且传感器感应到有人靠近给出高电平信号时，门才会打开，系统的真值表如下所示：

商店开门开关	顾客接近传感器	螺线管输出
Off	Off	Off
Off	On	Off
On	Off	Off
On	On	On

真值表是 AND 门逻辑，因此控制商店门的 PLC 梯形图如图 14-10 所示。



图 14-9　控制螺线管的开关



图 14-10　商店门系统

14.5　指令列表

在梯形图程序中的每个水平梯级代表了程序中的一行，因此输入一个个梯级就给出了“梯形图语言”的完整程序。程序员可用有图形化符号的键盘输入梯形图要素，或用计算机屏幕和鼠标选择符号，于是程序面板或计算机就把这些符号要素翻译成可以存储在 PLC 内存中的机器

语言。除了梯形图外，还有一种输入程序的方法就是把梯形图程序换化为指令表程序，然后把它们输入到编程面板或计算机中。

指令列表由一系列不同行的指令组成。一条指令由一个操作符和一个或者多个操作数（操作符的操作对象）组成。对梯形图的程序而言，程序中的每一个操作符可以视为一个梯形图要素。这样我们可以找到与梯形图中输入相等价的程序，例如：

```
LD A ( * Load input A* )
```

操作符 LD 用于载入，操作数 A 是被载入的操作对象，括号内 * 号之间的文字部分并不属于 PLC 程序的操作指令，而是起到帮助读者理解程序的注释作用。

不同的 PLC 制造厂商有不同的 PLC 助记码，但是提出了一个国际标准（IEC 1131-3），并得到了广泛应用。表 14-1 就是部分最核心的助记码。在本章其余部分讨论的例程中，如果没有专门的说明，我们将使用 Mitsubishi（三菱）助记码。然而，不同制造商的助记码的差异并不是非常显著，它们使用的原理也都基本相同。

表 14-1　指令助记码

IEC 1131-3	Mitsubishi	OMRON	Siemens	操　作	梯形图
LD	LD	LD	A	加载操作数到结果寄存器中	以常开触点开始一个梯级
LDN	LDI	LD NOT	AN	加载负操作数到结果寄存器中	以常闭触点开始一个梯级
AND	AND	AND	A	布尔与	与常开触点串联
ANDN	ANI	AND NOT	AN	与负操作数进行布尔与	与常闭触点串联
OR	OR	OR	O	布尔 OR	与常开触点并联
ORN	ORI	OR NOT	ON	与负操作数进行布尔或	与常闭触点并联
ST	OUT	OUT	=	将结果寄存器存储在操作数中	从一个梯级的输出

14.5.1　指令列表和逻辑函数

下面的部分将主要阐述梯形图的梯级程序如何用 Mitsubishi 助记符来输入，其中还包含一些逻辑函数（见图 14-11）。

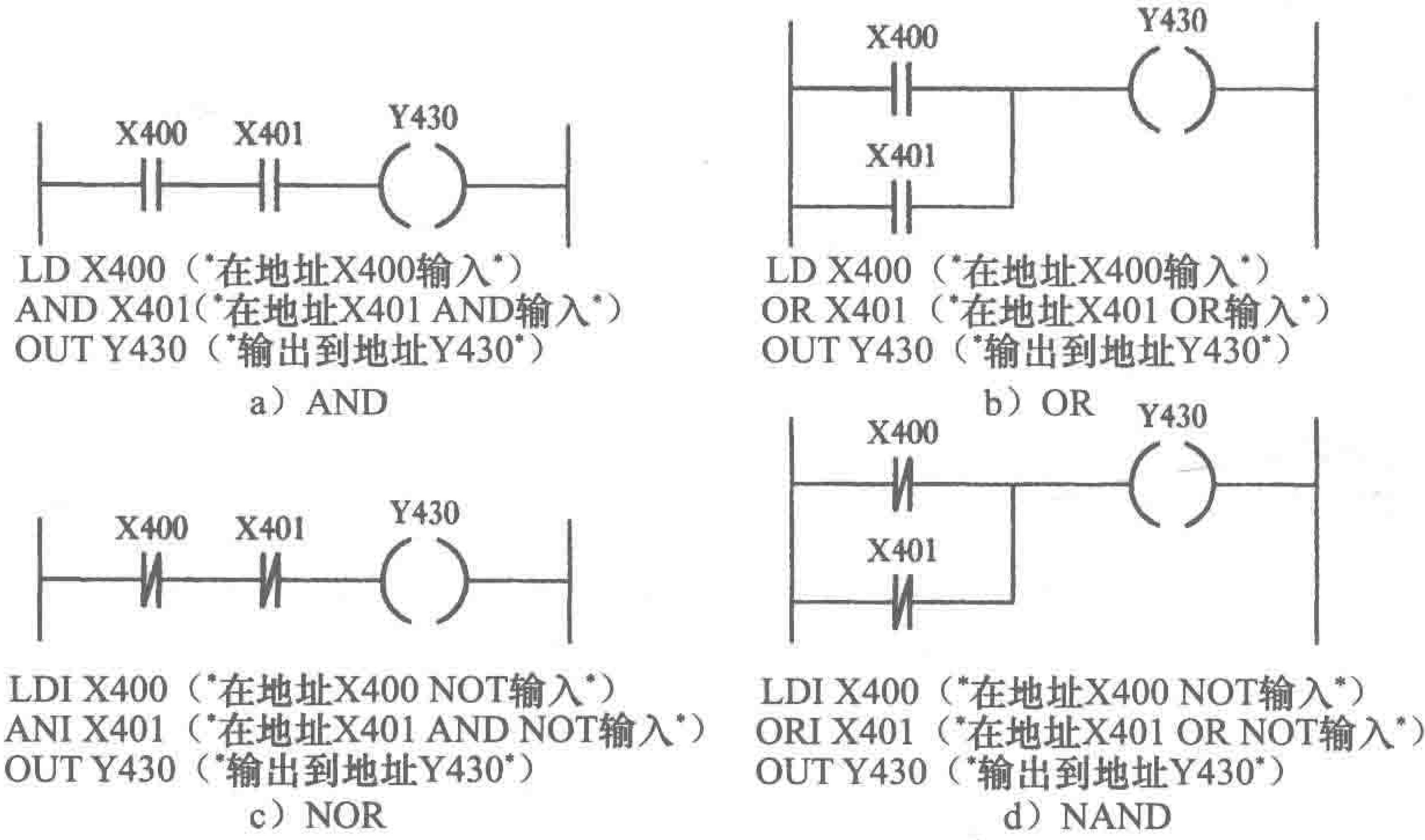


图 14-11　指令表和逻辑函数

14.5.2　指令列表和分支程序

图 14-12 中的异或门（XOR）有两个并联的分支，每个分支都用一个 AND 门连接。在这样的情况下，Mitsubishi（见图 14-12a）使用 ORB 指令来对所有的并联分支进行 OR。第一个指令表示一个常开触点 X400，下一条指令表示一个常闭触点 X401，因此有 ANI X401。第三条指令表示一个新的

行，理由是它由 LDI 开始，所有的新行都由 LD 或者 LDI 开始。因为第一行的末尾没有输出，所以 PLC 认为有第二条线与它并联，因此会一起读所列出的元素直到遇到 ORB 指令。ORB 表示第一、第二条指令的结果与第三、第四条指令的新分支进行 OR。图 14-12b 是 Siemens 异或门的表示方式。使用括号表示某些指令是以块的方式来执行，像任何数学方程中的括号一样来使用。例如，(1+2)/4 表示 1 必须与 2 相加后再除以 4。因此，对于 Siemens 的指令表，“A(” 意味着先载入指令 A，在括号完成后达到 “)”。IEC 1131-3 标准也使用了类似上述 Siemens 例程中括号的使用方法。

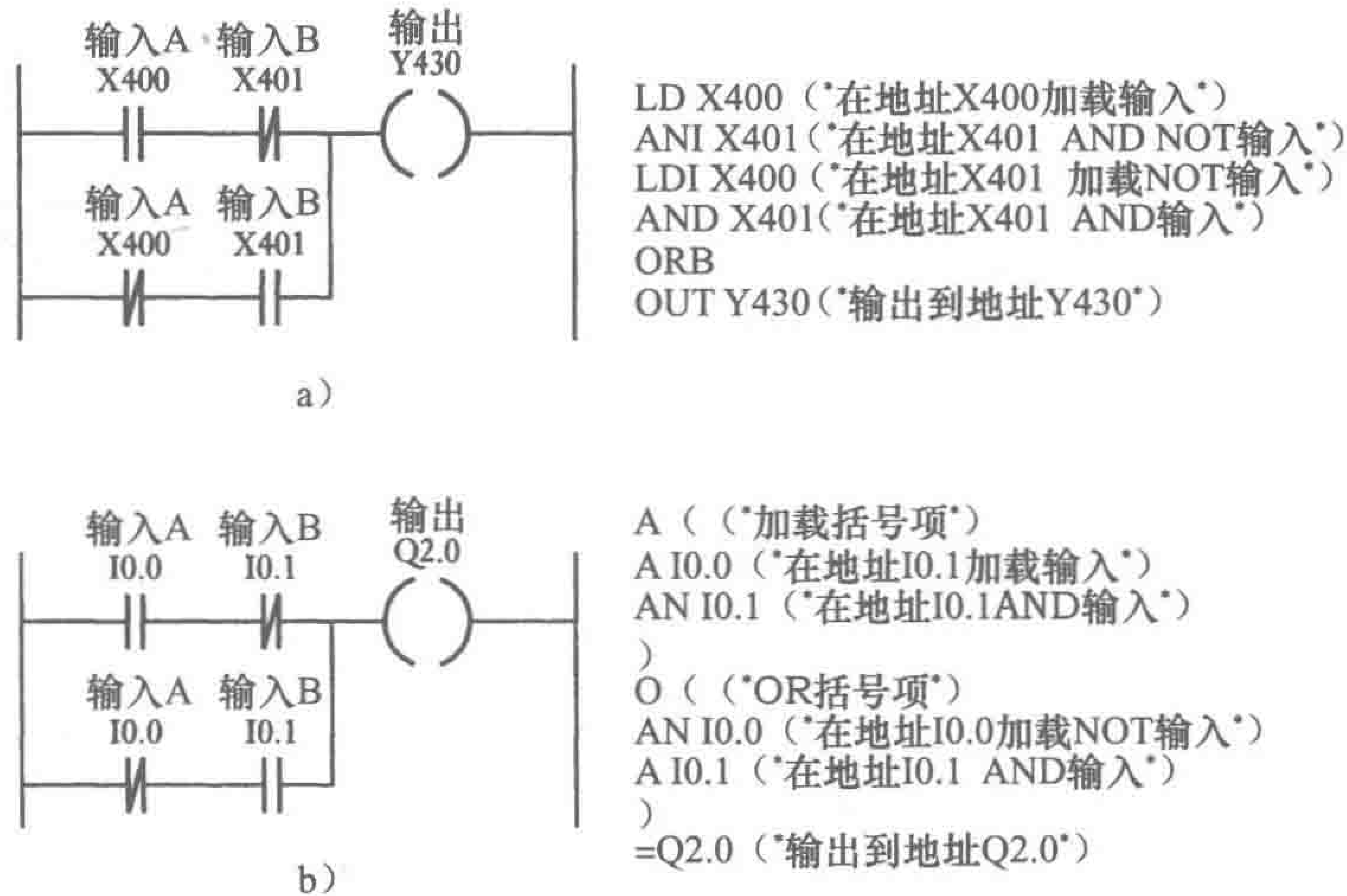


图 14-12　XOR

358～360

14.6　自锁和内部继电器

一般情况下，即使输入的信号停止，我们也需要保持线圈处于上电的状态。**自锁（或锁存）电路**就是要实现这样的操作。这是一个自保持电路，一旦启动，它的状态将保持直到下一个输入到来。图 14-13 就是一个这样的例子，当输入 1 闭合时，输出端有输出。这时，输出相对应的常开触点闭合，并与输入 1 的触点相 OR。这样，即使输入 1 的触点打开，电路会一直保持输出上电。唯一断开输出的方法就是接通常闭输入 2。

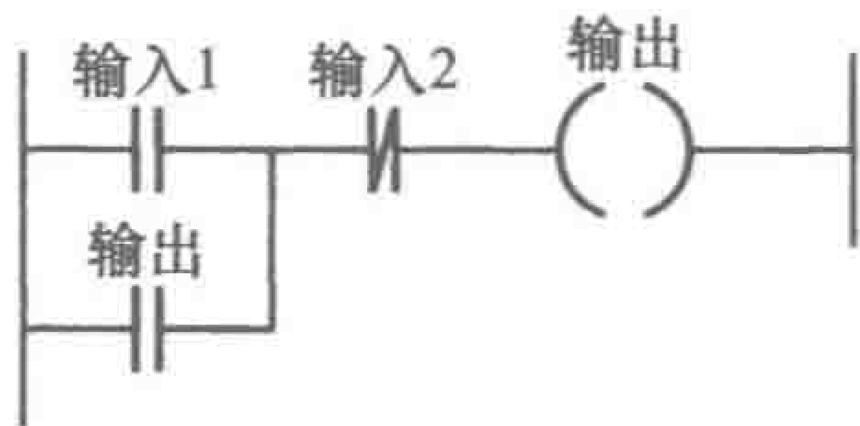


图 14-13　自锁电路

作为一个使用自锁电路的例子，考虑用一个 PLC 控制电动机，要求瞬间按下启动按钮时电动机启动，按下停止按钮时电动机停止。安全必须作为 PLC 系统设计的优先要素，因此停止按钮必须是硬线连接的而不能依赖于 PLC 软件来实现，因此当停止按钮或 PLC 发生故障，系统可自动保证安全。在 PLC 系统中，停止信号可以由图 14-14a 中的开关提供。开始的时候按下开始开关，这样电动机内部控制继电器的触点会形成自锁，因此输出能够持续供电。需要停止时按下停止按钮，这样会解除自锁并且停止电动机。但是，如果停止按钮出现故障无法操作，那么系统就无法停止，这样的系统不可采用，因为它不够安全。我们需要的系统必须能够在停止按钮失效的情况下仍能停止工作。梯形图程序现在让停止开关为常开的触点。图 14-12b 就是这样的一个系统。程序有一个常开触点的停止开关。然而，因为硬线停止开关用常闭触点，于是程序接收到关闭程序停止触点的信号。如果按下停止按钮，则打开程序触点，并停止系统。

14.6.1　内部继电器

术语**内部继电器**（internal relay）、**辅助继电器**（auxiliary relay），或者**标志**（marker）都可

以用做 PLC 的内部软继电器。这些继电器就像普通继电器一样控制相应的触点，但是它并不是真正的继电器，而功能的实现主要依赖于 PLC 软件的仿真。有些具有备用电池，因此即便在机器意外失效的情况下也能够保证机器安全关闭。内部辅助继电器在执行开关顺序的逻辑时非常有用。

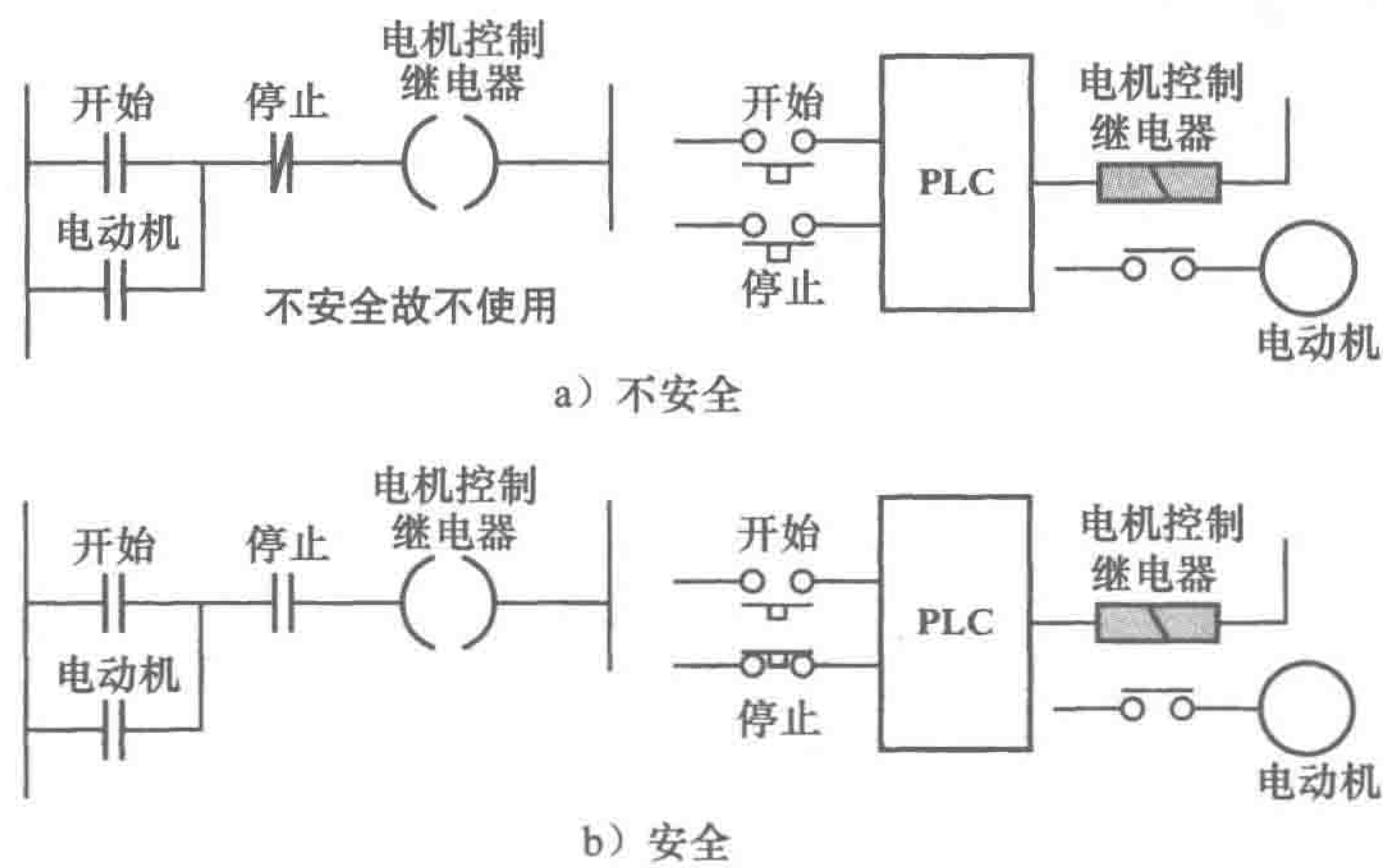


图 14-14 停止系统

内部继电器常用在有很多输入条件的程序中。考虑一种输出需要受到两种输入控制的 PLC 程序。图 14-15a 告诉我们如何用内部继电器来画梯形图。第一个梯级的输入控制 IR1 的线圈，第二个梯级的输入控制 IR2 的线圈，IR1、IR2 的触点以并联（OR）的形式控制输出。

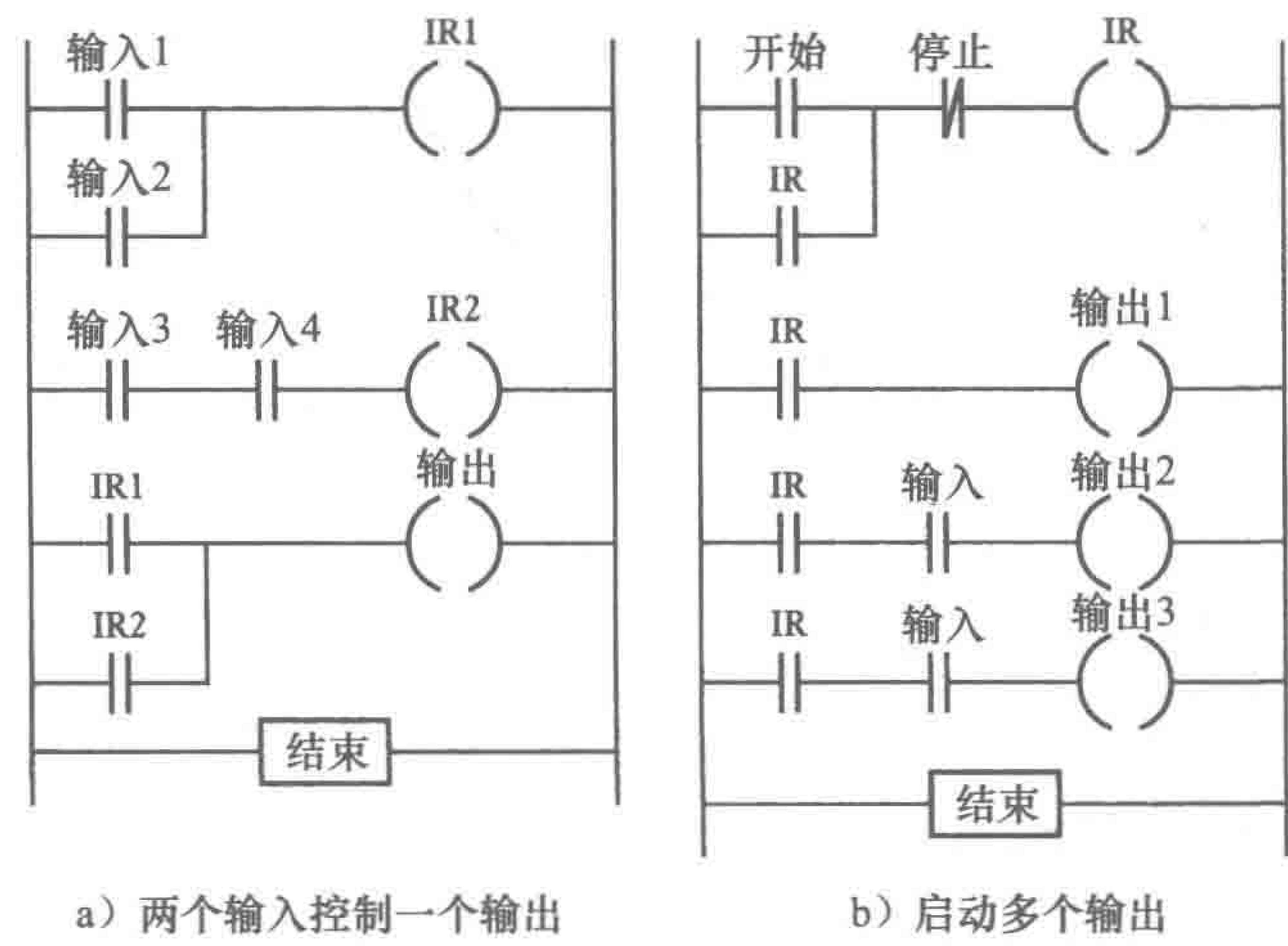


图 14-15 例子

继电器的另一个例子控制多个输出的启动。图 14-15b 就是这样的梯形图。当开始的触点闭合时，内部继电器被激活，IR 的所有触点都会闭合。输出 1 启动，并且也为输出 2 和输出 3 的启动提供了条件。

图 14-16 是一个运用内部继电器解除自锁的梯形图。当输入 1 这个触点按下时，输出线圈会上电。输出的触点闭合并且形成自锁，即使输入触点打开，输出也保持闭合状态。输出触点在输入 2 触点闭合时解锁。

图 14-17 是一个使用备用电池的内部继电器的例子。当输入的触点关闭时，备用电池的内

部继电器线圈会上电。这样，即使出现电源失效导致输入触点打开，内部继电器的触点仍然能够保持闭合，使输出仍然被内部继电器所控制。

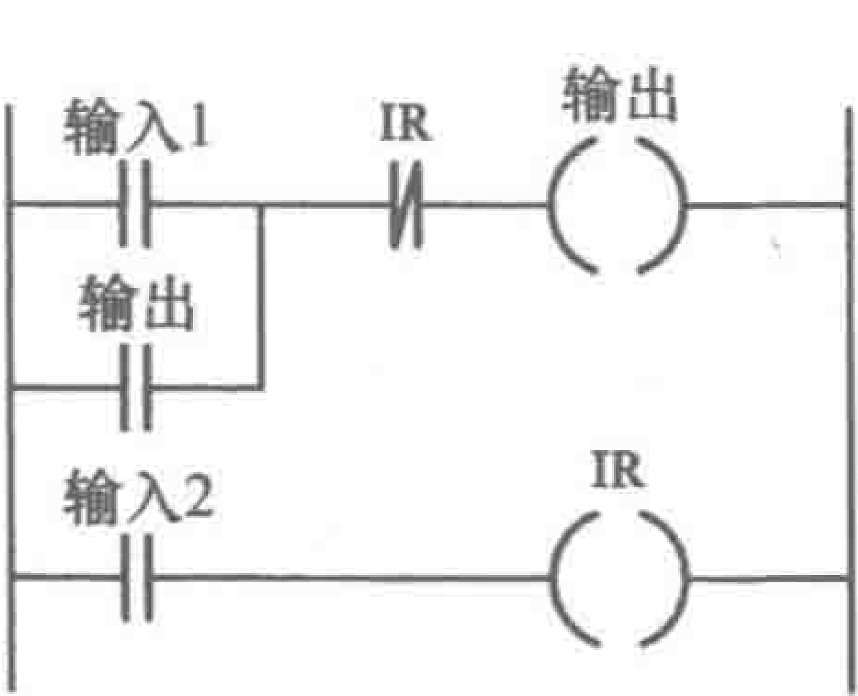


图 14-16　复位自锁电路

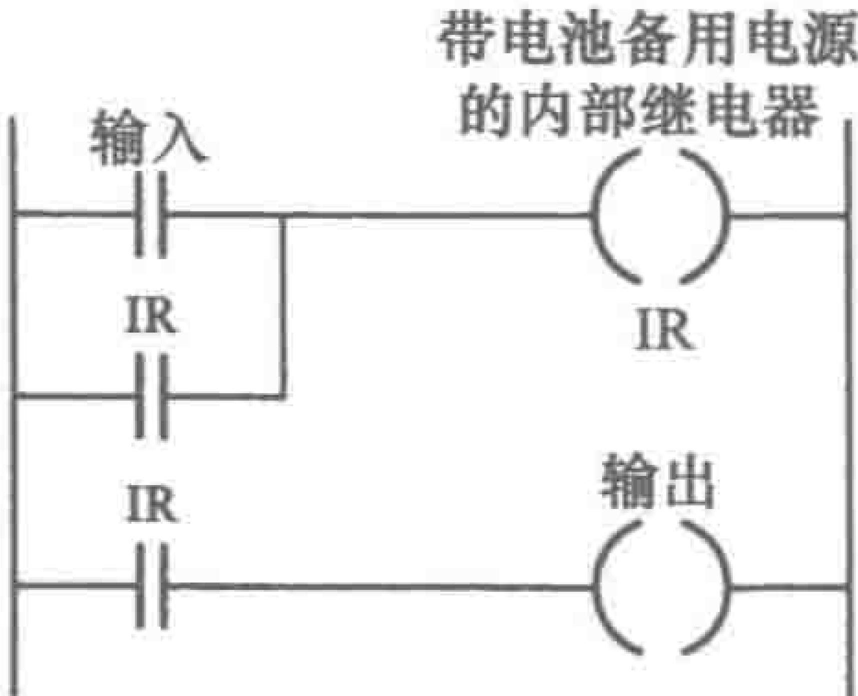


图 14-17　带电池备用电源内部继电器的使用

14.7　顺序控制

有些情况下输出需要顺序启动，从一个输出的开关到另一个输出的开关由传感器来控制。考虑一个能够满足以下气动系统控制要求的梯形图（见图 14-18）：用双螺线管电磁阀控制双作用汽缸 A、B，限位开关 a-、a+、b-、b+用来探测活塞杆在缸体内的极限位置，并且要求按 A+、B+、A-、B-的顺序工作。一个可行的梯形图如图 14-18 所示，梯形图的第一行有启动触点。这样，当且仅当开始按钮被按下，并且限位开关 b-也闭合时（表示 B 已被缩回），A+输出才能上电。当汽缸 A 伸出时，开关 a+有信号，这样输出 B+才能上电，使汽缸 B 动作，以此类推，可以顺序启动输出 A+、B+、A-、B-。程序运行到 B-处时执行完毕，回到最初的第一行等待开始按钮再次按下。

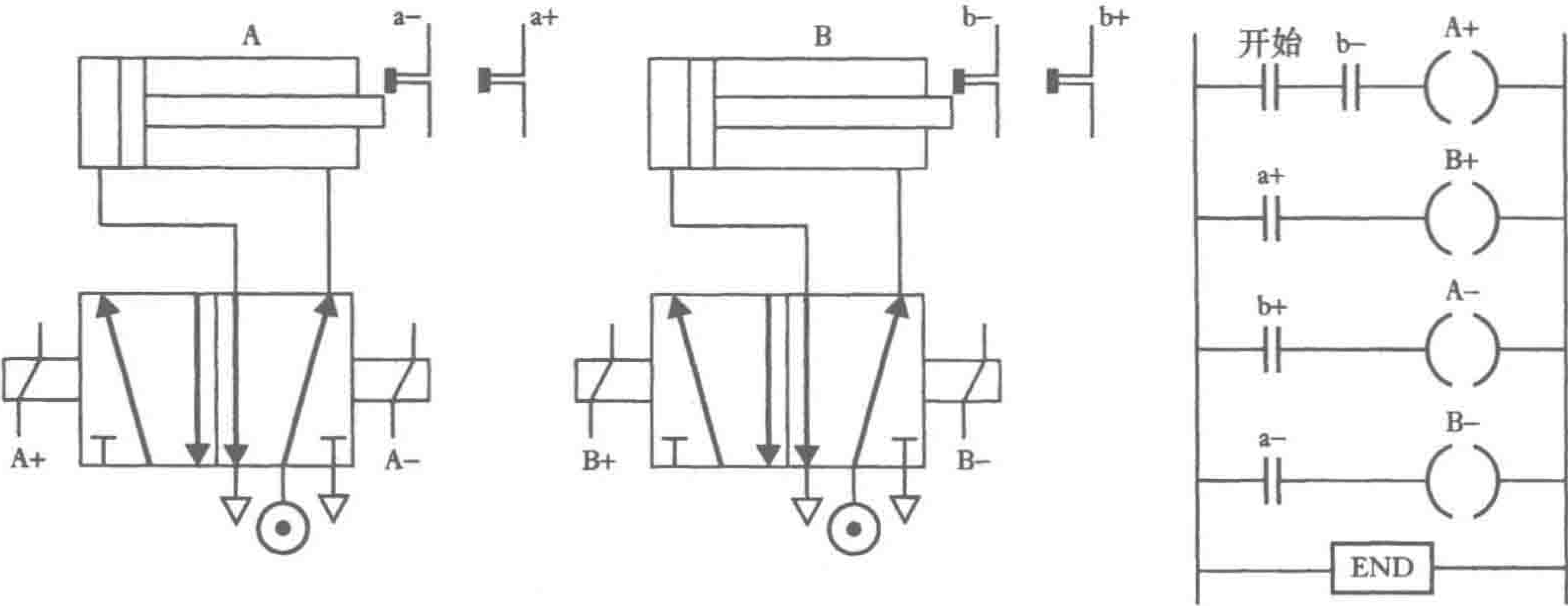


图 14-18　汽缸顺序控制

为了进一步阐述，我们考虑利用梯形图设计一个具有双螺线管控制阀的气动系统的问题。A、B 为两个汽缸，限位开关 a-、a+、b-、b+用来探测活塞杆的极限位置，工作顺序如下：A 活塞伸出，之后 B 中的活塞伸出，其次 B 中的活塞回缩，最后 A 中的活塞回缩。一个内部继电器可用于一组输出之间的开关，从而给出一种汽缸的控制形式，称为级联控制（见 7.5 节）。图 14-19 就是一种可行的梯形图。当按下启动按钮时内部继电器触点被激活。A+螺线管内部得电，使得 A 中的活塞伸出，当它靠近限位开关 a+时，B 中的活塞开始伸出。当前进至限位开关 b+时内部继电器会被激活，这时螺线管 B-得电，B 中的活塞回缩，当靠近限位开关 b-时，螺线管 A-得电，汽缸 A 中的活塞回缩。

361
~
363

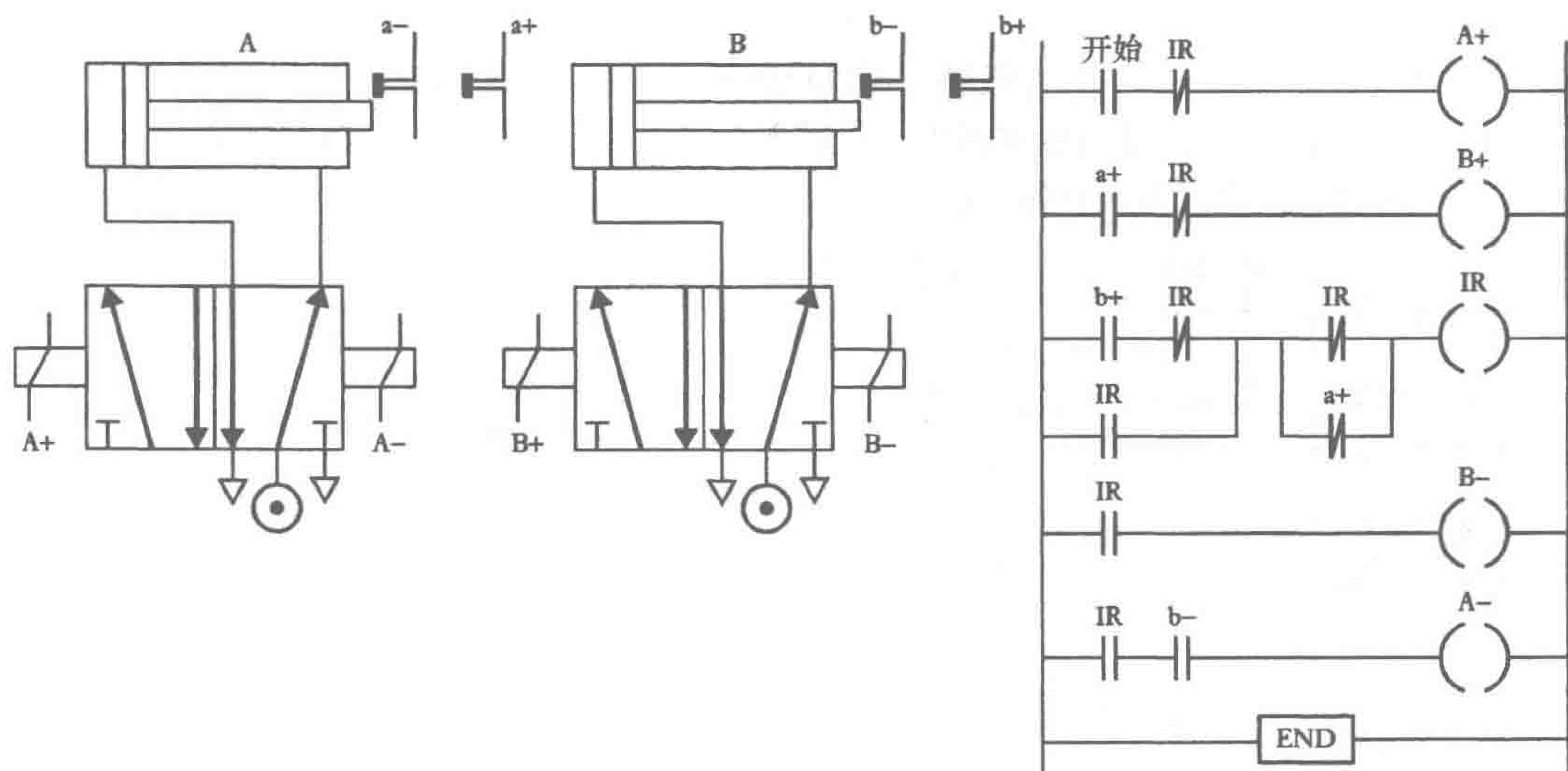


图 14-19　汽缸顺序控制

14.8　定时器与计数器

本章的前面章节主要讨论了需要串联和并联输入触点来实现控制任务的问题。然而，有很多任务都需要延时与计数。这些需求可以通过 PLC 的定时器与计数器来实现，这是 PLC 的一个特征。它们可由逻辑指令控制，并在梯形图上表示出来。

14.8.1　定时器

PLC 制造商的常用方法是让定时器像有线圈的继电器一样，在上电后经过一段预置的延时时间触点才开始关闭或打开。因此定时器的触点可以作为梯形图中执行控制的某行的输出被用在其他地方（见图 14-20a）。除此以外，也可以把定时器看作某一行中输入对输出控制的延时模块（见图 14-20b）。

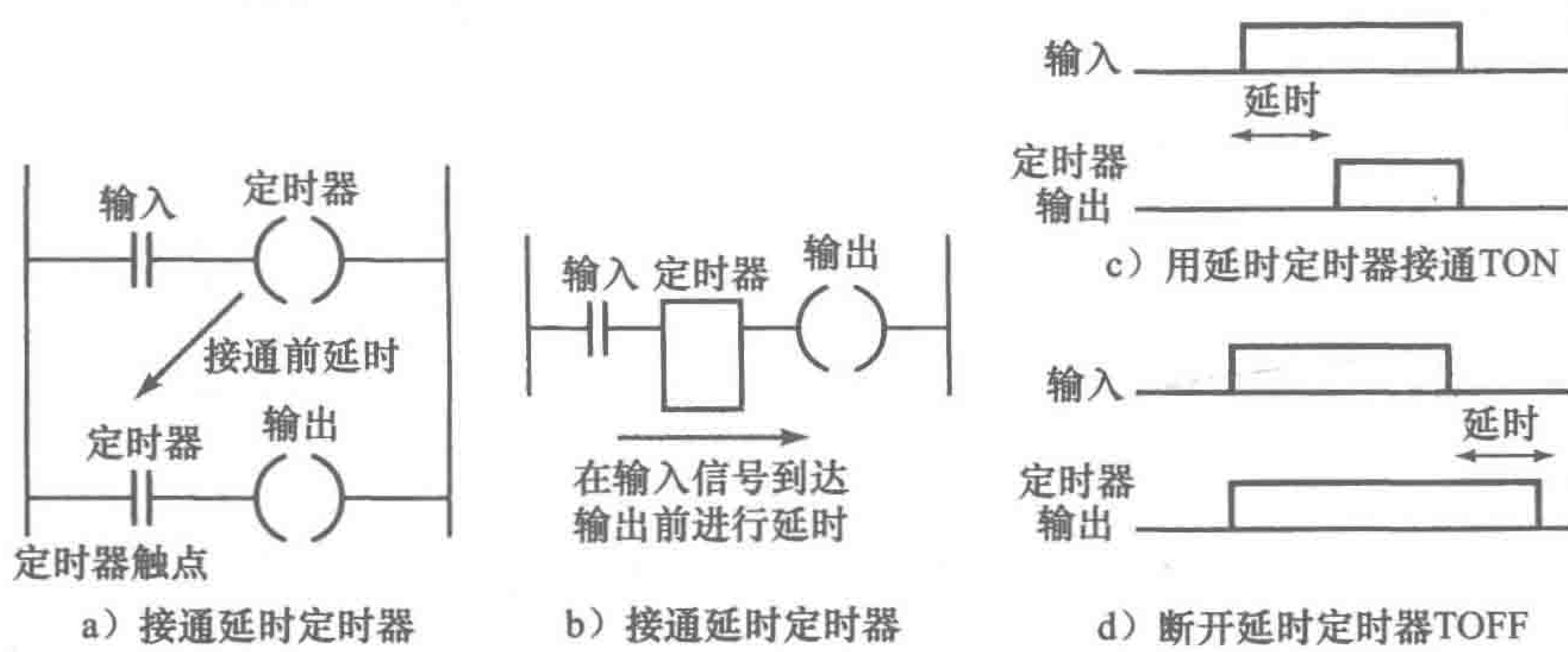


图 14-20　定时器

PLC 通常提供延时启动定时器（TON），小的 PLC 大多只有这种模块。这样的定时器在打开前会等待一个固定的时间（见图 14-20c），例如这个时间段通常在 0.1 ~ 999s 之间，间隔步长为 0.1s。当然也可能有其他的时间延迟范围和间隔步长。

为了说明如何使用定时器实现顺序控制，考虑图 14-21a、b 的梯形图。当输入 In1 接通时，输出 Out1 接通。那么与输出相关的触点启动定时器。当预先设定的延时到时，定时器触点闭

合，输出 Out2 接通。

定时器可以连接在一起，或者**级联**，可以得到相对于一个定时器而言更大的定时时间。图 14-22 就是这样的程序。当输入触点通电，定时器 1 启动。当定时器 1 定时时间到时，它的触点闭合，定时器 2 启动。当定时器 2 定时时间到时，它的触点闭合，输出线圈接通。

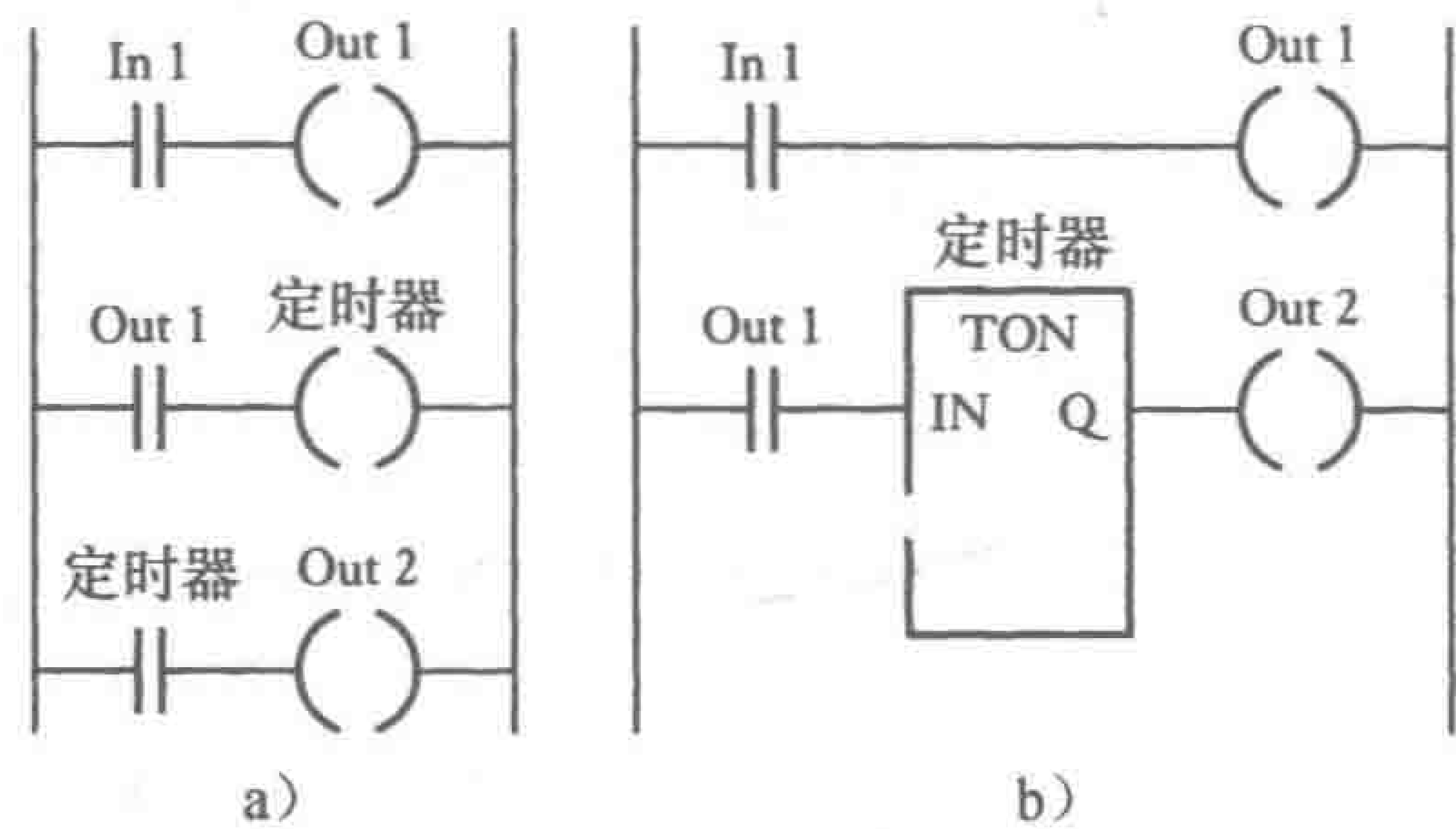


图 14-21　定时序列

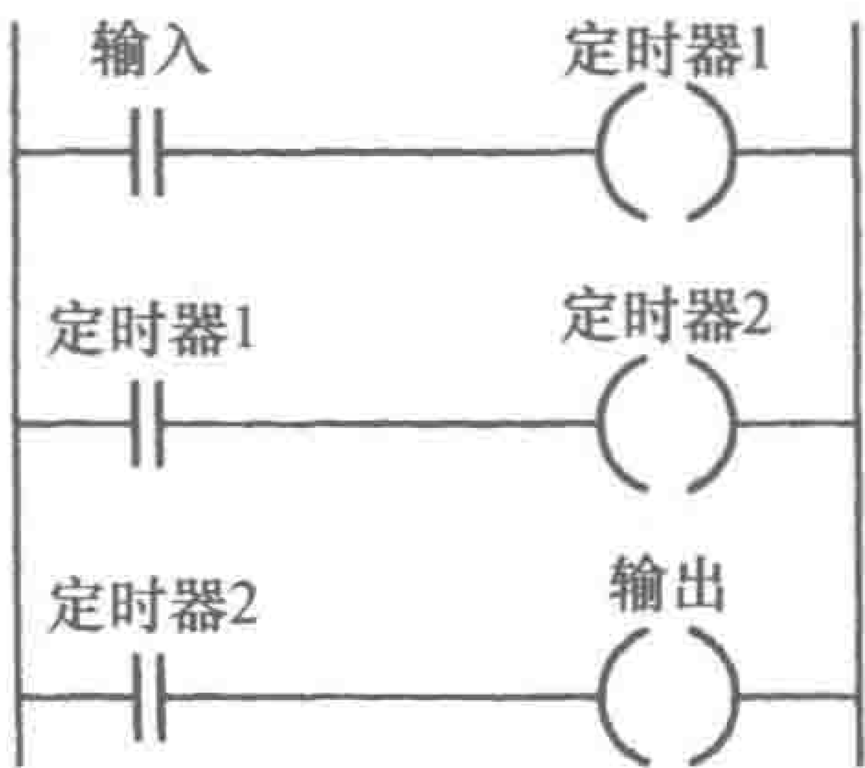


图 14-22　级联定时器

图 14-23 是一个能够实现 0.5s 闪烁的梯形图，输出端能够工作 0.5s，停止 0.5s，如此往复。工作原理如下：当输入触点闭合时，定时器 1 开始计时，0.5s 之后，第二行的定时器 1 的触点会接通，定时器 2 开始计时，同时，第三行的定时器 1 的触点也会被接通，输出端开始接通。0.5s 后定时器 2 定时时间到，第一行的定时器 2 的常闭触点断开，定时器 1 重新计时，定时器 1 的触点打开使得定时器 2 断电，另外第三行的输出端也会相应地断开。系统开始往复循环。这个结果是定时器 1 的触点接通 0.5s，关闭 0.5s，再接通 0.5s 等，使输出也进行同样的动作。

图 14-24 是一个延时断开的定时器。当输入触点短暂接通时输出端开始接通，在输入常闭触点断开时定时器开始计时（见图 14-20d）。输出触点并联到输入触点（自锁），保持输出。当定时时间到时，定时器会断开自己的常闭触点和自锁电路，使输出断开。

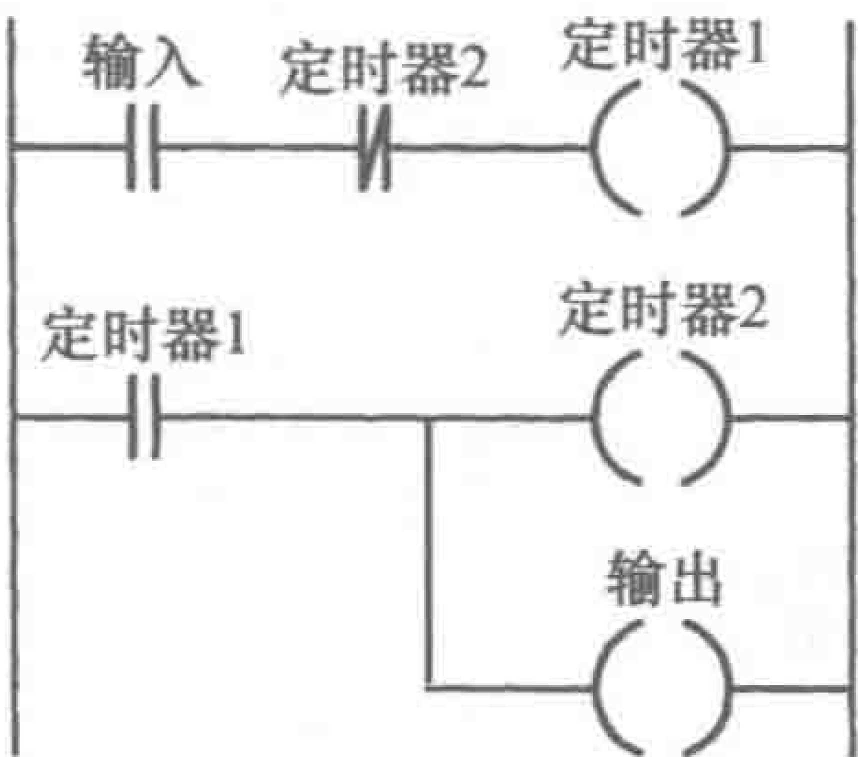


图 14-23　开/关周期定时器

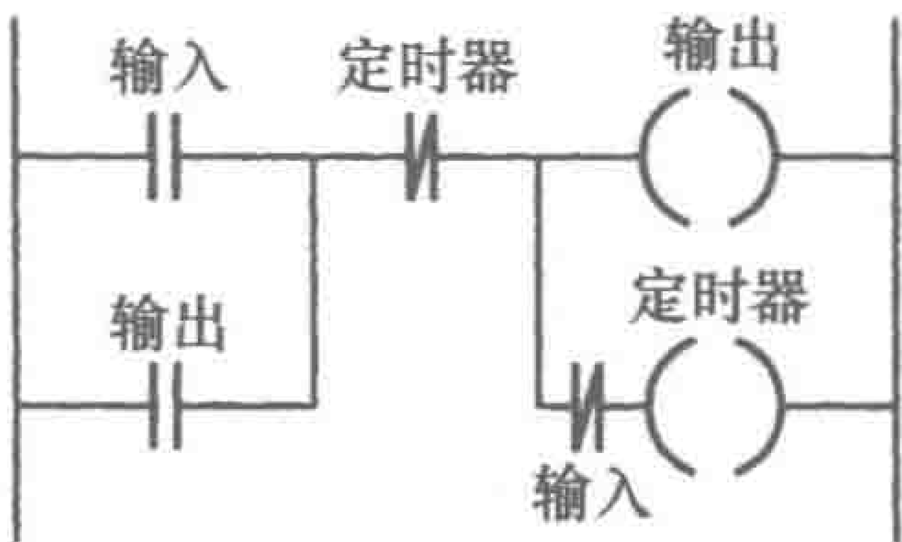


图 14-24　延时断开的定时器

14.8.2　计数器

364
~
365

当需要对某一操作进行计数时，会用到计数器。例如，当我们需要把沿着传送带移动的物品送入不同的箱子里，当指定数目的物品已送入一个箱子后，下一个物品就要分配到其他箱子里。计数器电路是 PLC 提供的内部特征，在大多数情况下计数器以**减计数器**（down-counter）的方式工作。这意味着计数器从当前某一个特定的数值减到 0，当减到 0 时计数器的触点改变状态。另外还有**加计数器**（up-counter），当加到设定值时计数器的触点改变状态。

不同的 PLC 厂商处理计数器的方法不同。有的厂商（Mitsubishi 和 Allen Bradley）认为计数器有两个基本组成元素，一个是用来对输入脉冲计数的输出线圈，另一个用来复位计数器。计数器对应的触点可用在梯形图的其他行。还有的厂商，如 Siemens，把计数器当作一个计满数时立即产生一个信号的单元。图 14-25 就是一个基本的计数电路。当输入 1 中有脉冲信号输

入时，计数器复位，当 In2 中有脉冲信号时，计数器开始计数。假设计数器的设置值为 10，那么当 In2 中收到 10 个脉冲信号时，计数器的触点将会接通，Out1 端将会有输出。如果在计数的任意时刻计数器收到 In1 信号，那么计数器将复位，并重新开始计数到 10 个脉冲。

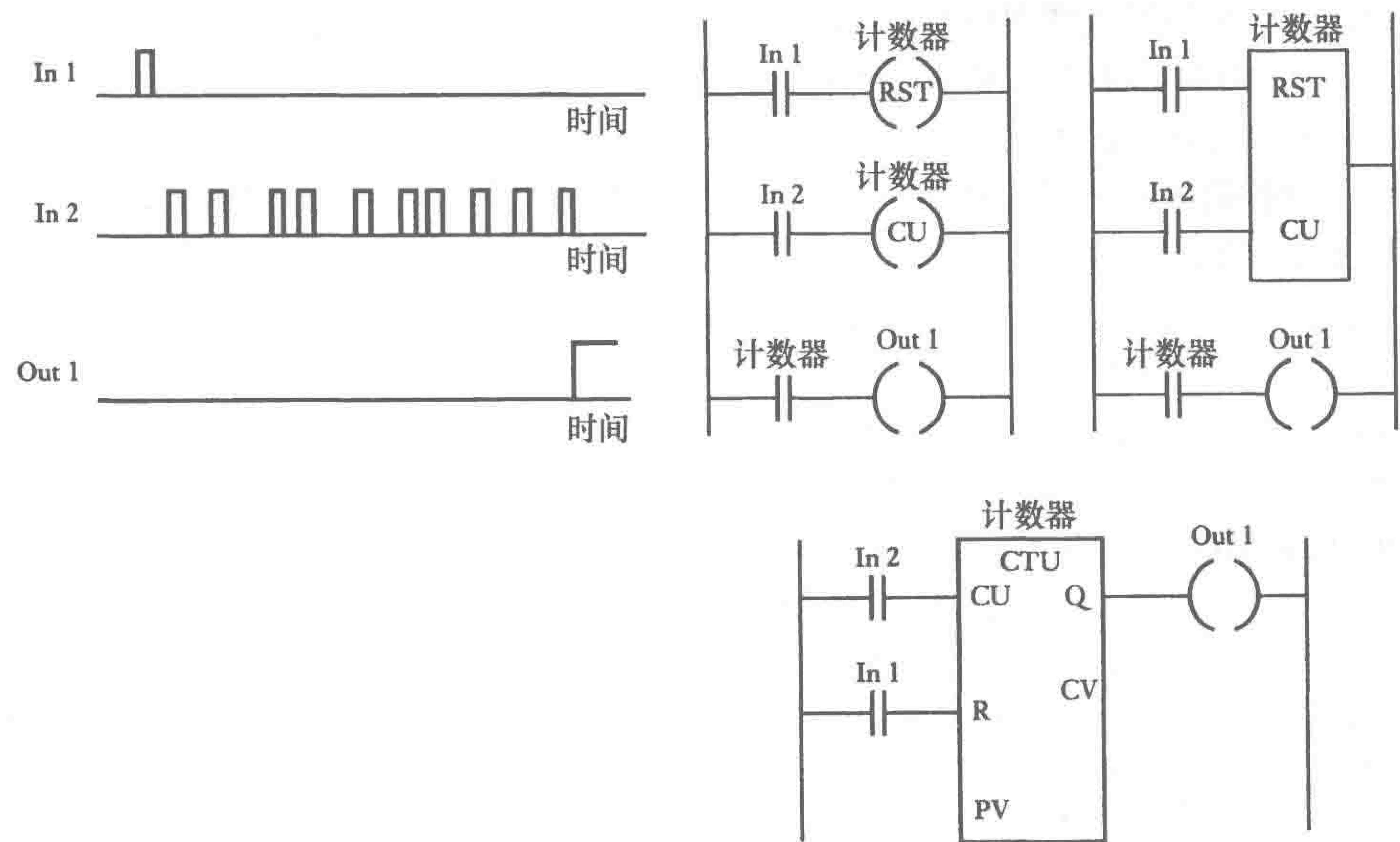


图 14-25　计数器的输入和输出以及同一程序的不同实现方式

为了说明计数器的应用，考虑沿着一条轨迹在一个盒子内包装 6 个商品，然后沿着另一条轨迹在另一个盒子内包装 12 个物品的问题。图 14-26 就是一个运用计数器的程序。包含 2 个计数器，一个计数值预置为 6，另一个预置为 12。当输入 1 瞬间闭合触点时，计数循环开始，并复位两个计数器。输入 2 的触点可以用一个微动开关激励，每当一个商品经过时都会产生一个信号。这样，当计数器 1 计满 6 时，计数器 1 的触点闭合。这会产生一个输出，可用一个线圈控制一个薄片关闭一条路径，并且打开另一条路径。同时计数器 1 也有触点关闭并使能计数器 2 来开始计数。当计数器 2 计数满 12 时，它的触点能够复位两个计数器，并打开计数器 1 的触点，停止输出，使商品不进入包含 12 个物品的盒子。

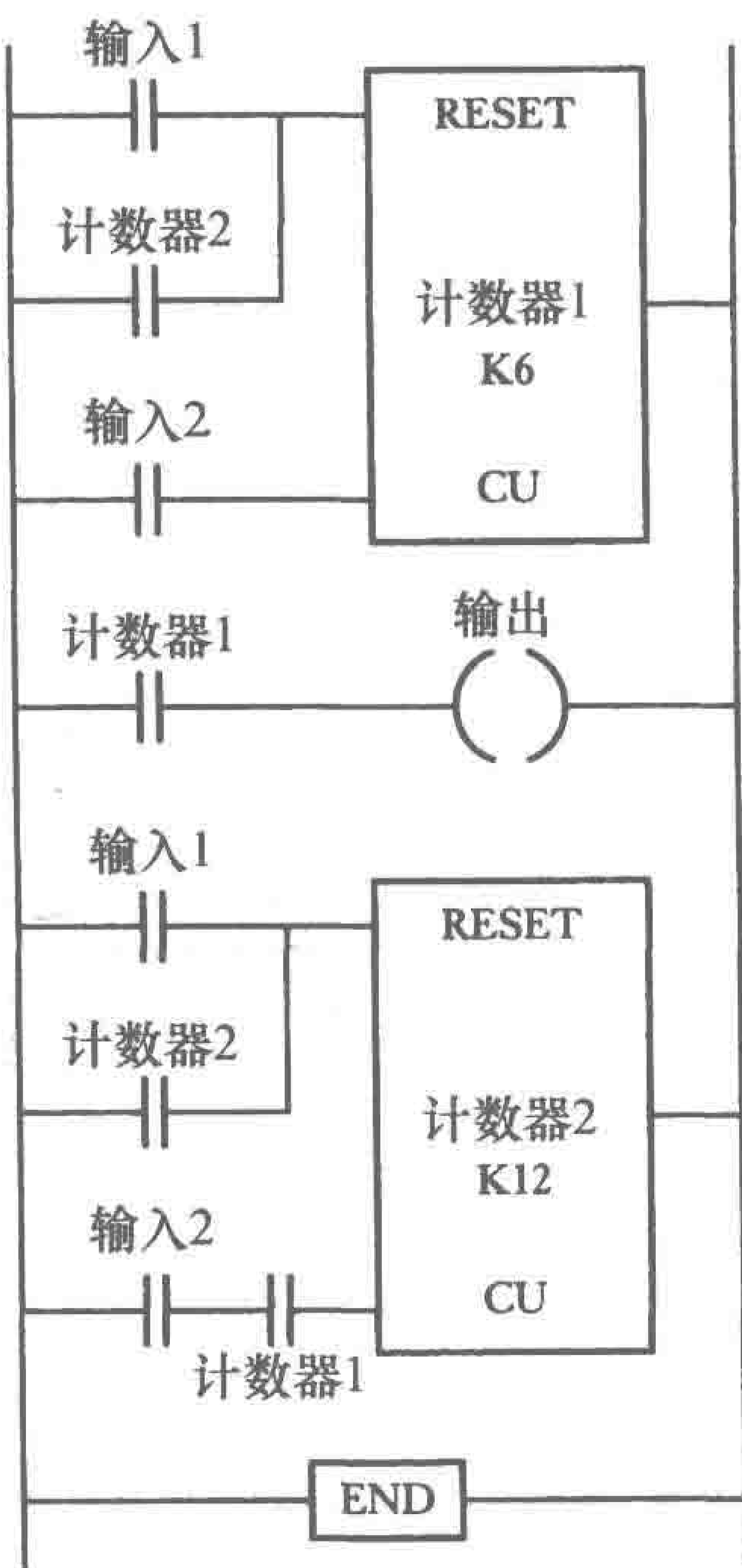


图 14-26　计数器

366

14.9　移位寄存器

多个内部继电器能够组合成一个寄存器，这个寄存器可以为一系列的位提供存储空间。一个 4 位的移位寄存器需要 4 个内部继电器，一个 8 位的寄存器需要 8 个内部继电器。使用**移位寄存器**（shift register）这个术语是因为当寄存器有合适的输入时，寄存器中的位移动一位。举个例子，对于 8 位的寄存器有初始值为：

1	0	1	1	0	1	0	1

当左边有一个 0 移位脉冲输入时，

0 →

0	1	0	1	1	0	1	0

→ 1

每一位都相应地移动了一位，并且原始值最末端的一位会溢出。

当在控制面板中选择移位寄存器函数时，PLC 自动地将一组辅助寄存器转换为移位寄存器。对于 Misubishi PLC，这是通过用编程函数 SFT（移位）和辅助继电器号（该号码是寄存器阵列中的第一个继电器）实现的。于是从该号码开始，引出了一串内部继电器保留用于移位寄存器。因此，如果我们选择 M140 作为第一个继电器，那么该移位寄存器将会由 M140、M141、M142、M143、M144、M145、M146、M147 组成。

移位寄存器有 3 个输入，第一部分将数据加载到寄存器的第一个输出端（OUT），第二部分是移位命令控制的输入端（SFT），第三部分是复位端（RST）。对于 OUT，逻辑电平 0 或者 1 会加载到移位寄存器的第一个部分。对于 SFT，一个脉冲信号会将寄存器中的所有内容相应地移动 1 位，最后的 1 位会溢出，并丢失。对于 RST，只要收到一个脉冲信号，那么移位寄存器中的所有内容都会被清零。

图 14-27 是一个包含 Mitsubishi 移位寄存器的梯形图，其他制造商的 PLC 的原理类似。M140 指定为移位寄存器的第一个继电器。当 X400 接通时，逻辑 1 载入到移位寄存器的第一个单元，也就是 M140。这样，寄存器变为 10000000。电路图中表示了移位寄存器的每一个单元都作为触点被连入电路。这样，M140 的触点闭合，Y430 也相应地接通。因为 X400 依然是接通的，所以当触点 X401 接通时，寄存器移位后变为 11000000，一个 1 被移入寄存器。因此，M141 的触点闭合，Y431 也相应地接通。当每一位移动时，相应的输出也依次接通。这样，移位寄存器可以用做事件的顺序控制。

367

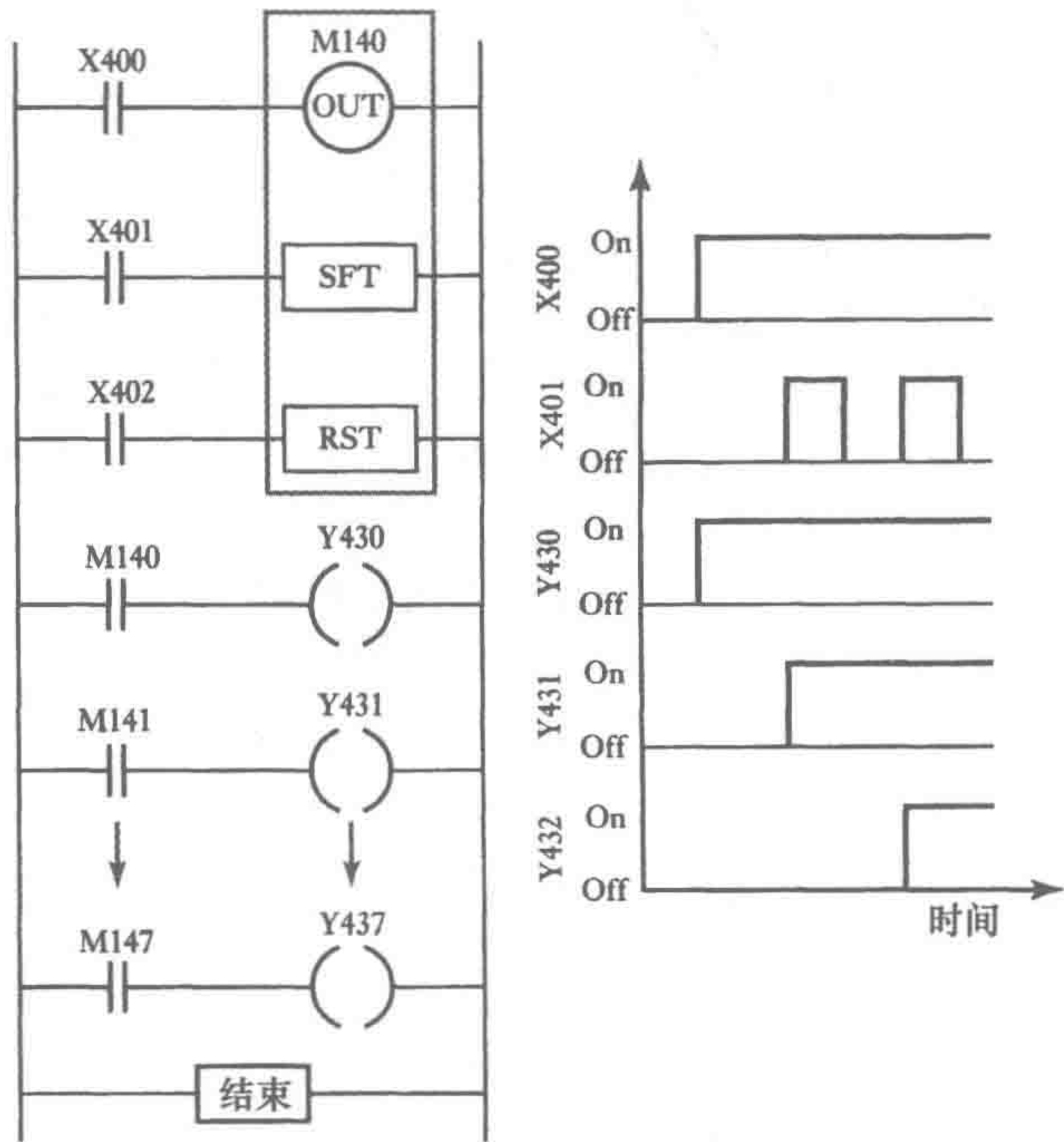


图 14-27　移位寄存器

14.10　主控指令和跳转指令控制

多个输出可以同时打开或者关闭，只要控制它们的触点是同一个。因此，通过打开或关闭一个继电器，就会影响到有相关触点的其他程序行。一个相同的控制方法就是使用**主控继电器**（master relay）。图 14-28 就是它的一种用法的说明。我们可以认为它的触点用于控制垂直轨道的电源，当输入 1 接通的时候，MC1 线圈通电，对应的触点 MC1 接通，这样后

面的一段程序行就被主控继电器所控制。另外，主控继电器的结束由最后一行的复位 MCR1 表示。当输入 1 中有输入时，程序沿着 MC1 的控制路线执行；如果没有，就执行后面的程序，跳过这个分支。

对于 Mitsubishi PLC，可以在程序中将内部辅助继电器编程指定为主控继电器。因此，将内部继电器 M100 变成主控继电器的程序指令为：

```
MC M100
```

表示主控继电器所控制部分结束的指令为：

```
MCR M100
```

14.10.1 跳转

PLC 经常提供的一个功能是条件跳转功能。这样设计的目的在于一旦某个条件满足，那么程序可以转到相应的地方执行。图 14-29 就是这样的一个流程图和梯形图的说明。在程序 A 之后，梯形图会遇到输入 1 和条件跳转继电器 CJP。如果输入 1 有输入，那么程序直接跳转到有结束跳转的继电器 EJP，并继续执行 C，而不会执行 B；如果输入 1 中没有输入，那么程序将按照 A、B、C 的顺序执行。

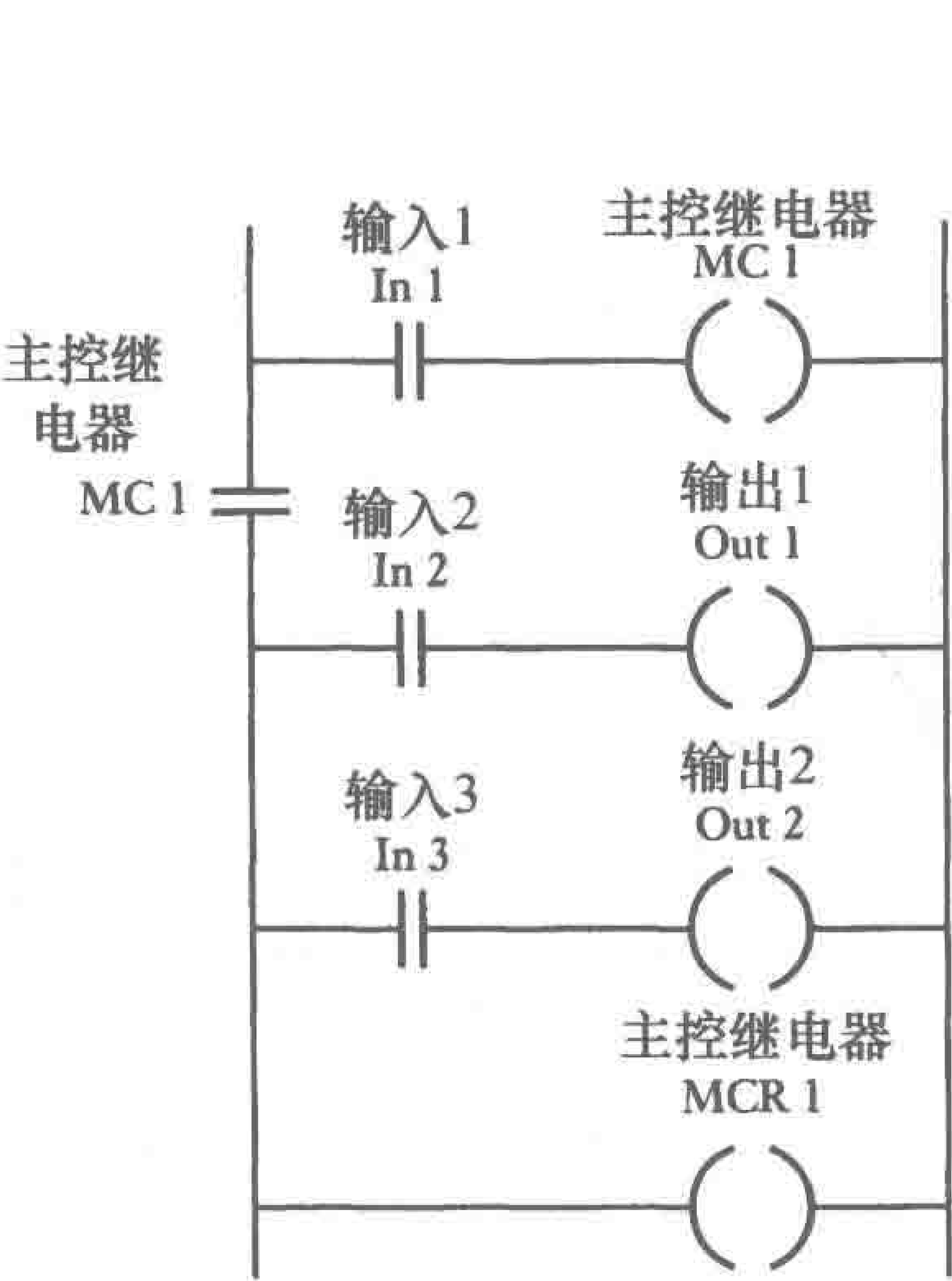


图 14-28　主控继电器

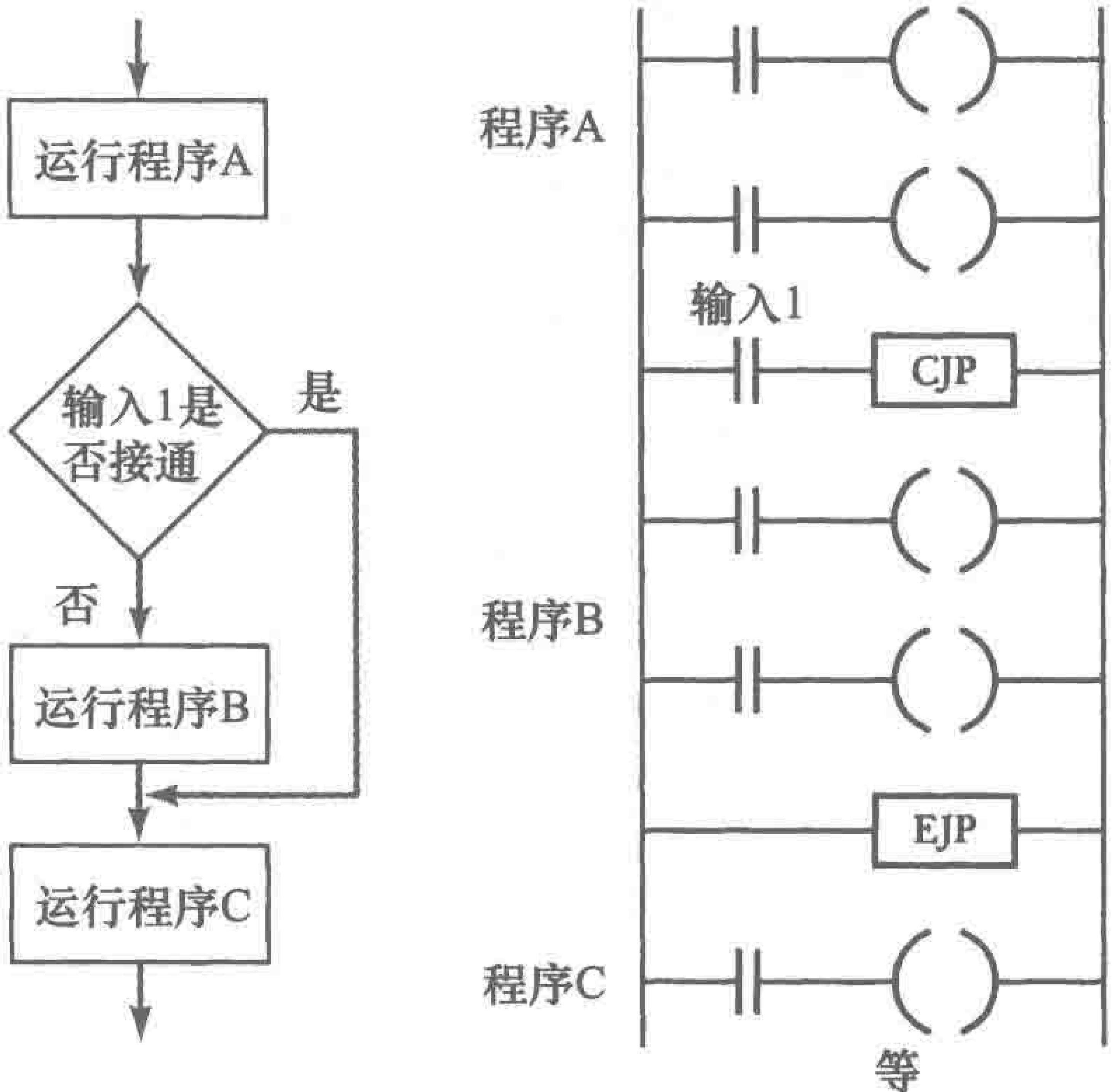


图 14-29　跳转

14.11 数据处理

除了移位寄存器外，本章前面的部分都只是考虑单独的位的信息处理，例如，一个开关的开或闭。然而，在有些情况下，操作一组位会非常有用。例如，可以把 8 个输入点当作一个 8 位的数据字节来处理。这样的情况会出现在以下的情形中：例如，在输入到 PLC 前，传感器可将模拟信号转换为有一个 8 位的字节。

368
~
369

可以通过 PLC 进行的数据字操作有：

1） 移动数据。

2） 比较数据值的大小，即大于、等于或小于。

3） 算术运算，例如加和减。

4） BCD 码、二进制和八进制之间的转换。

正如之前所描述的那样，单独的位存储在唯一地址指定的存储器区域内。例如，对于

Mitsubishi PLC，输入内存地址都以 A 开头，输出都以 Y 开头，定时器都以 T 开头，辅助继电器都以 M 开头。PLC 也需要特定的空间存储数据指令，用来存储数据的 PLC 存储器的空间称为**数据寄存器**（data register）。每个数据寄存器可以存储一个类似于地址 D0、D1、D2 等的 8 或者 16 位的二进制数。8 位字的精度是 1/256，16 位字的精度是 1/65 536。

每条指令都应该指定操作方式、源数据寄存器和目标数据寄存器。

14.11.1　数据移动

对于数据移动，需要包含数据移动指令、数据源地址，以及数据的目的地址。这样的梯形图具有图 14-30 的形式。

这样的数据传送可能是将一个常量输入到数据寄存器中，也可能是将一个时间值和计数值传送到数据寄存器中，也可能是将数据寄存器中的数据传送到定时器与计数器中，或将数据寄存器中的数据传送到输出，输入数据传送到数据寄存器等。

14.11.2　数据比较

PLC 通常可以使用以下形式的数据比较，例如小于（记作 < 或者 LES）、等于（记作 = 或者 EQU）、小于等于（记作 ≤ 或 <= 或 LEQ）、大于（记作 > 或 GRT）、大于等于（记作 ≥ 或 >= 或 GEQ）、不等于（记作 ≠ 或 <> 或 NEQ）。数据比较程序包含比较指令、数据的源地址和目的地址。图 14-31 梯形图的意义是：比较数据寄存器 D1 中的数据是否大于数据寄存器 D2 中的数据。

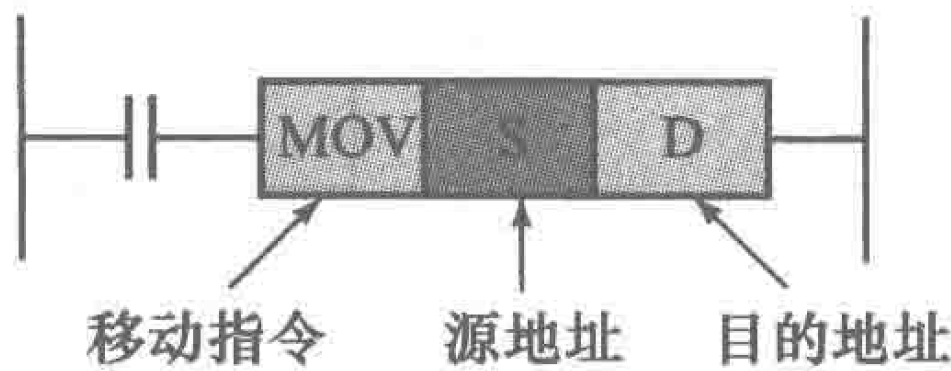


图 14-30　移动数据

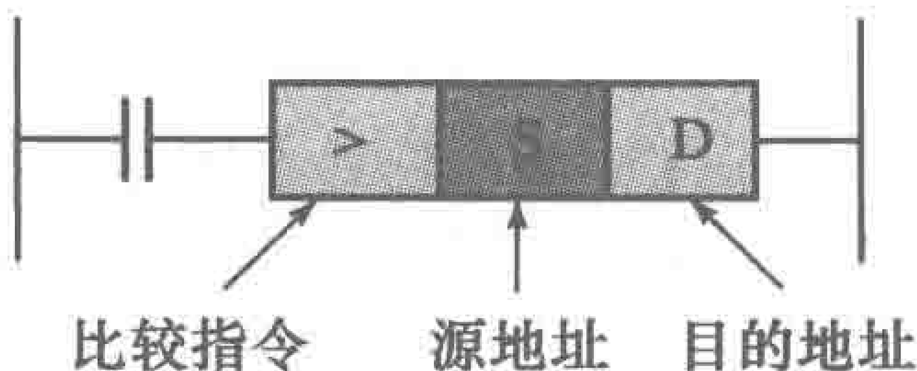


图 14-31　比较数据

这样的比较可能会运用在以下的情形中，在执行特定行动之前，通过 PLC 比较来自两个传感器的信号大小。例如，一个温度报警器需要在温度高于 80℃ 的时候发声报警，报警要求一直持续到温度低于 70℃。图 14-32 就是这样的一个梯形图。将输入温度数据输入到源地址，目的地址包含设定值。当温度上升到 80℃ 或更高时，源地址中的数据 ≥ 目的地址中的数据，报警线圈加电并自锁，报警器工作；当温度等于或低于 70℃ 时，源地址中的数据 ≤ 目的地址中的数据，IR 线圈加电，IR 的常闭触点断开，这样报警线圈掉电，报警器停止工作。

370

14.11.3　算术运算

有些 PLC 只包含加运算和减运算，另一些 PLC 具有更加强大的运算功能。通常需要声明运算指令，包含需要被加或者被减的数据地址的寄存器，以及存储输出结果的寄存器。图 14-33 给出了 OMRON PLC 的加法梯形图的形式。

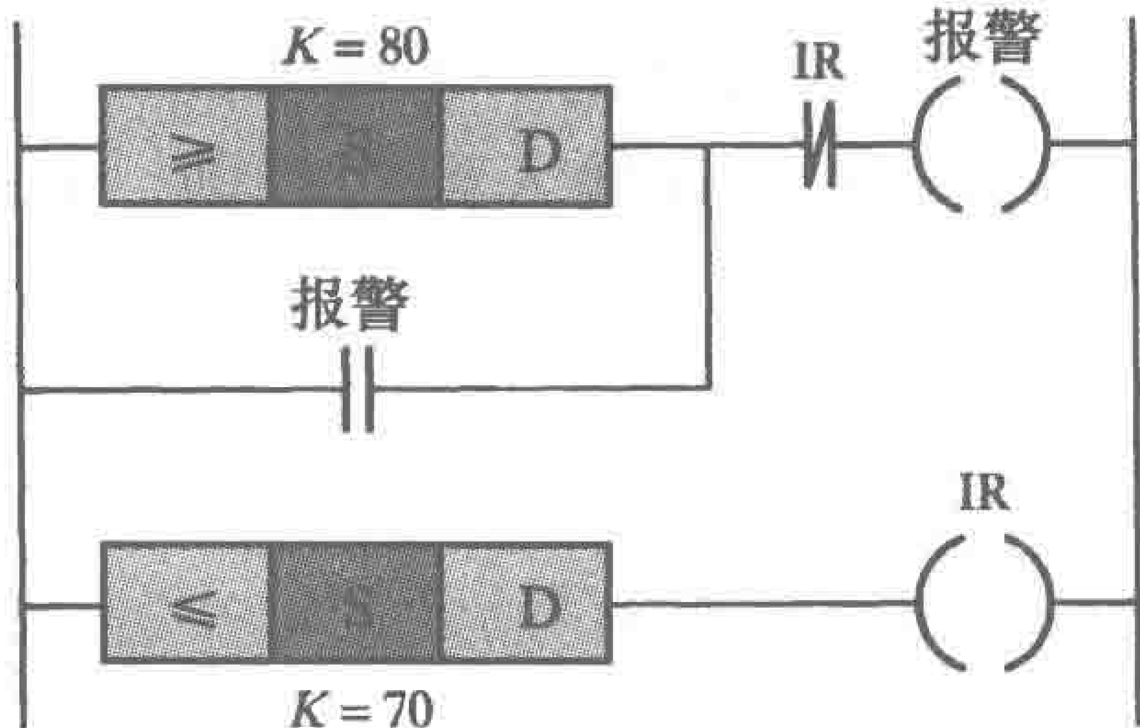


图 14-32　温度报警

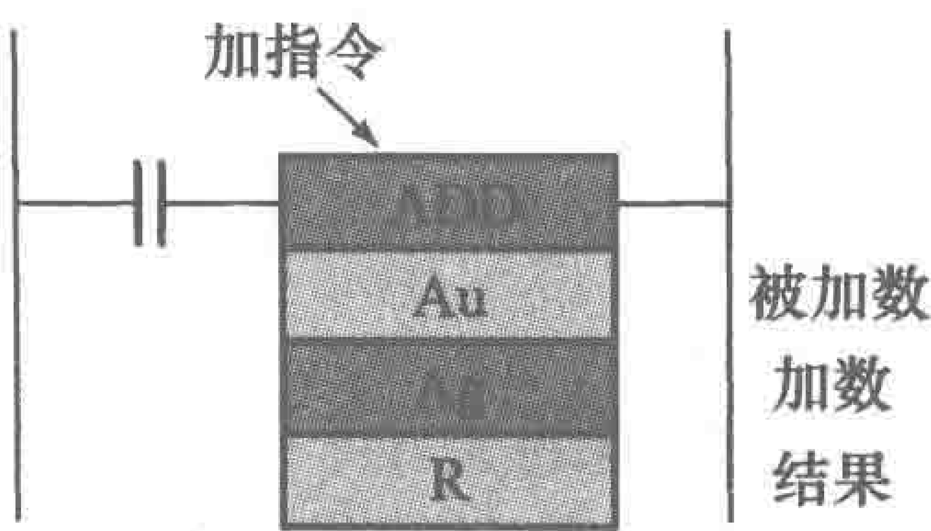


图 14-33　加数据

加法或减法运算可以用于改变一些传感器的输入数值（可能是修正和偏置项），或者修改

定时器或计数器的预置值。

14.11.4 代码转换

在 PLC CPU 中的所有内部操作都是通过二进制实现的。因此，当输入是十进制信号时，需要转换为 BCD 码。类似地，如果需要十进制输出，那么就需要到十进制的转换。大多数 PLC 都提供这样的转换功能。例如，图 14-34 就是 Mitsubishi BCD 码转换成二进制的梯形图的形式。源地址中的数据是 BCD 码，被转换成二进制并放置在目的地址中。

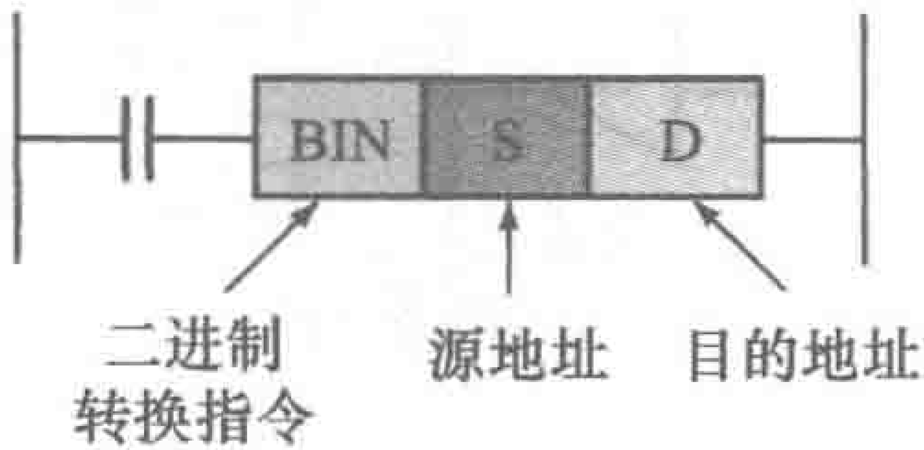


图 14-34 BCD 码到二进制转换

14.12 模拟输入/输出

许多传感器产生模拟信号，并且许多驱动器需要模拟信号。因此，有些 PLC 在输入端口有模数转换器（ADC）模块，输出端口有数字量转为模拟量的数模转换器（DAC）模块。这样的功能可以用做电动机转速的控制，例如，我们希望电动机能够以稳定的增速增加到稳定转速（见图 14-35）。输入由开关控制，当开关按下时它打开相应的常闭触点，这样数据寄存器的初始值就是 0。因此，控制器的输出也是 0，然后从 DAC 输出的模拟量也是 0，电动机没有转速。按下开始按钮时，开始的常开触点闭合，向 DAC 以及数据寄存器输出，每一个程序循环数据寄存器自加 1，因此输出的模拟量也会相应地增加，从而带动电动机稳定加速。当数据寄存器的值达到 11111111 时，电动机转速达到峰值。可以使用一个定时器，这样在每个输出位信号之间可以延长一定的时间。

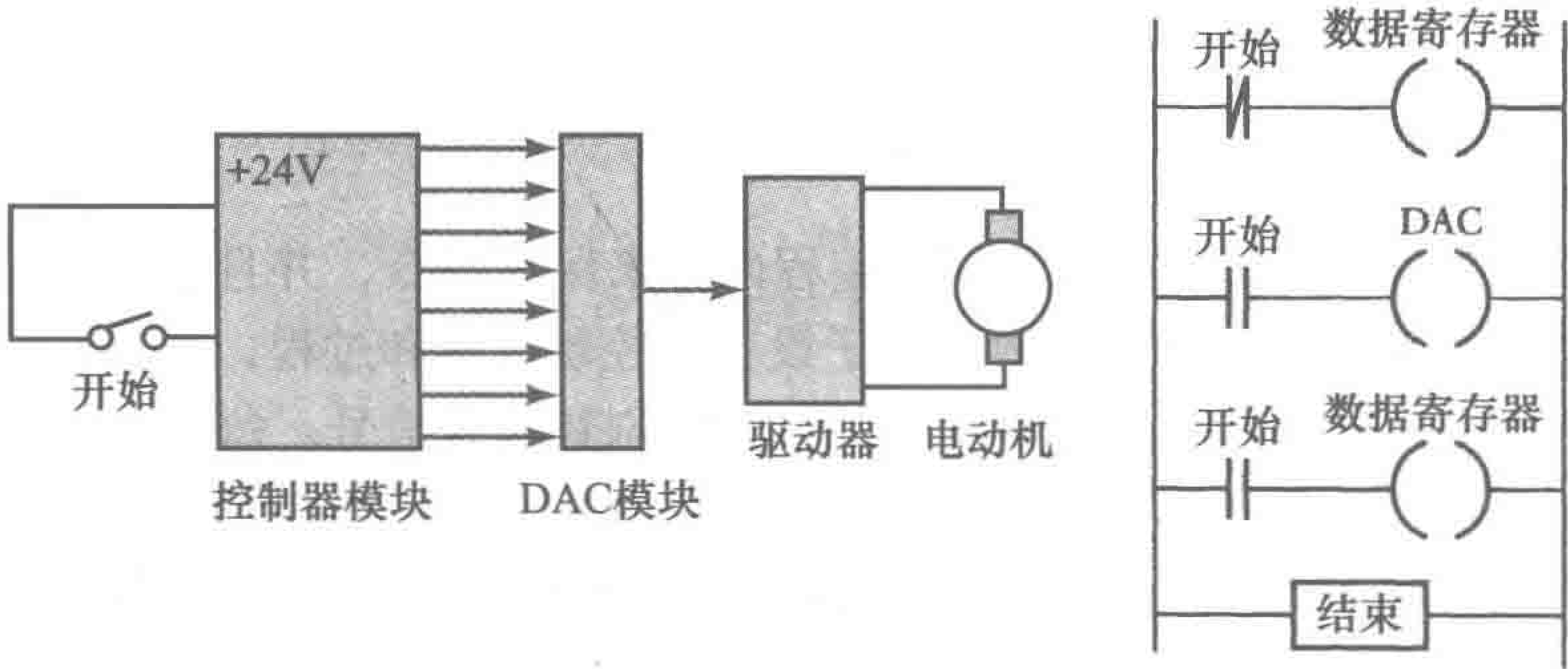


图 14-35 使电动机转速斜坡上升

带有模拟输入通道的 PLC 可以实现连续控制功能，例如 PID 控制（见 22.7 节）。这样，我们可以通过以下的指令来实现模拟输入的比例控制：

1）把传感器输出转换成数字信号。

2）将实际传感器转换成的数字输出与要求的传感器的值比较，得到误差。

3）将误差乘以比例常量 Kp。

4）把这个结果移入 DAC 输出，并且使用这个结果作为驱动器的修正信号。

下面是一个温控系统，可能会用到上述的比例控制（见图 14-36）。输入可能来自热电偶，放大后通过 ADC 进入到 PLC。PLC 通过编程给出与传感器输入和需要温度之间的误差成比例地输出。输出字通过 DAC 送入驱动器，即加热器，来减小误差。

如梯形图所示，第 0 行读 ADC 的数值并将温度值存入数据寄存器 DR1。在第 1 行中，数据寄存器 DR2 存储设定温度。第 2 行运用减法将这两个值做减法运算，并且将结果送入数据寄存器 DR3。第 3 行使用乘法运算，将 DR3 中的数值放大比例增益 4。第 4 行使用内部辅助继电器

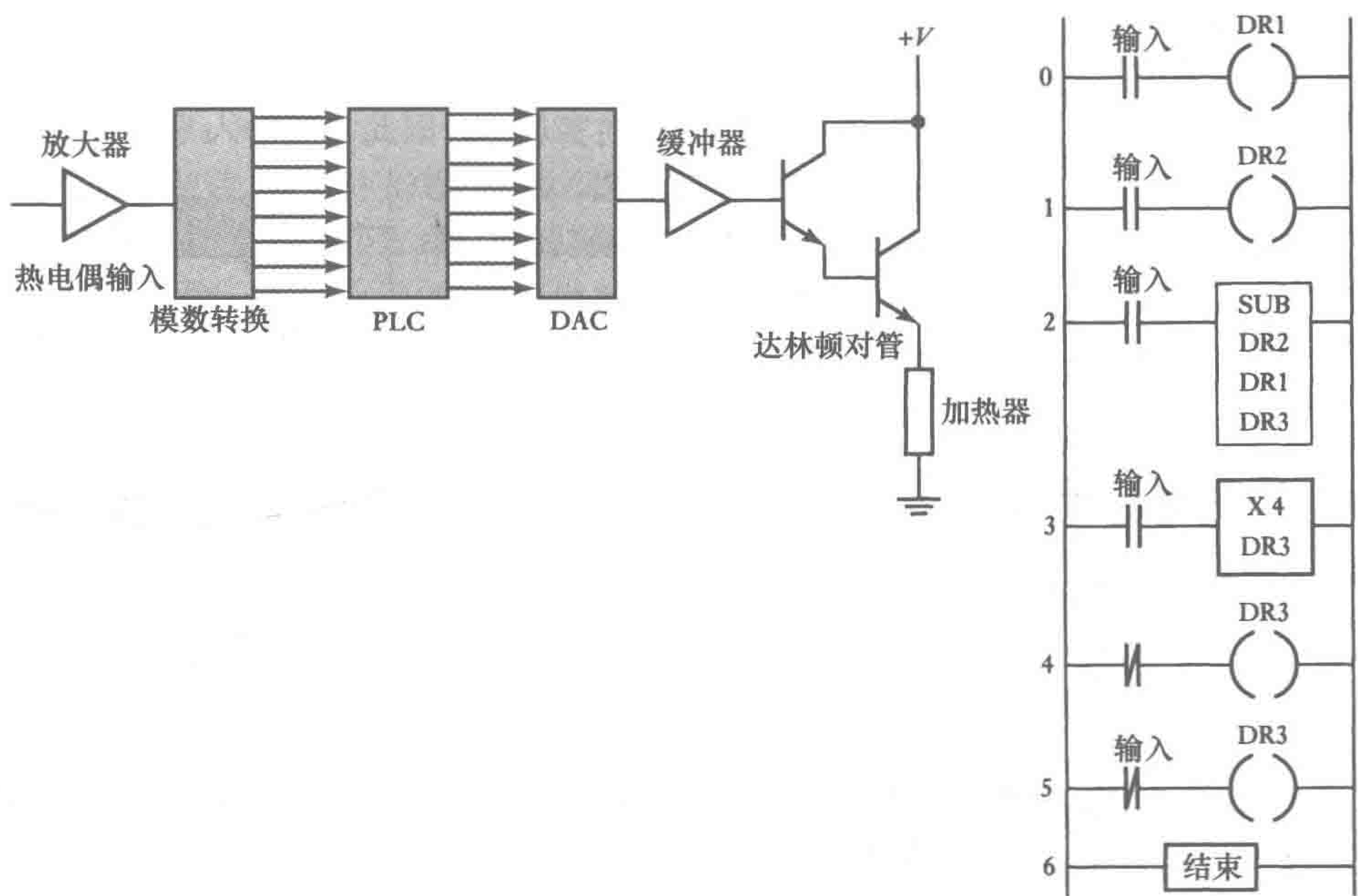


图 14-36 温度的比例控制

在 DR3 为负的情况下断开 DR3。如果输入开关打开，第 5 行数据寄存器 DR3 就清零。有的 PLC 具有扩展模块，使得 PLC 能够很容易地实现控制，而不用像上面一样写一系列的指令。

本章小结

可编程逻辑控制器（PLC）是一个用可编程内存来保存指令，并且能够执行逻辑、顺序控制、定时、计数和算术运算等功能的数字电子设备，控制机器和过程，并且使编程简化。

PLC 工作时不停地从头到尾执行程序，并且不断刷新输入信号。每一个这样的循环称为**周期**。PLC 常用的编程方法是**梯形图编程**。将每一个程序任务指定为梯形图中的一行。而另一种等同的输入程序的方法就是把梯形图翻译成**指令列表**的编程方法。指令列表由一系列指令组成，每一条指令独立占一行。一条指令由一个操作符和一个或者多个操作数（操作的对象）组成。

一个**自锁电路（或锁存电路）**就是即使输入开关断开，输出的状态也能够保持，直到收到下一个断开输入。术语**内部继电器**（internal relay）、**辅助继电器**（auxiliary relay），或者**标志**（marker）都可用做 PLC 的内部软继电器。这些继电器就像普通继电器一样控制相应的触点。**定时器**与有线圈的继电器一样，在上电后经过一段预置的延时时间，触点才开始关闭或打开，作为输入与输出信号的延时。**计数器**用于对触点操作进行计数，它有对输入脉冲进行计数的输出线圈，还可以对计数器进行复位，计数器的触点可用于其他的行，当计数达到一定数值时发出信号。**移位寄存器**就是由一些内部辅助继电器构成的一系列位的寄存器。**主控继电器**可以使整块程序运行或者停止。**条件跳转指令**是当特定条件满足时程序跳到相应的位置执行。以**数据字**的形式执行的操作通常包括移动数据，数据大小比较，算术运算以及 BCD 码、二进制和八进制之间的数码转换。

372 ~ 373

习题

14.1 下列开关的逻辑功能是什么：（a）串联；（b）并联。

14.2　请绘制梯形图来表示：

(a) 两个用于电动机操作的开关是常开的，且两者必须都闭合时电动机才能运行。

(b) 两个常开开关，两个开关必须闭合才能使线圈通电并操作驱动器。

(c) 电动机通过按压弹簧复位启动开关按钮来接通，电动机保持在接通状态，直到另一个弹簧复位停止开关按钮被按下。

14.3　请编写图 14-37 所示的自锁程序相对应的指令。

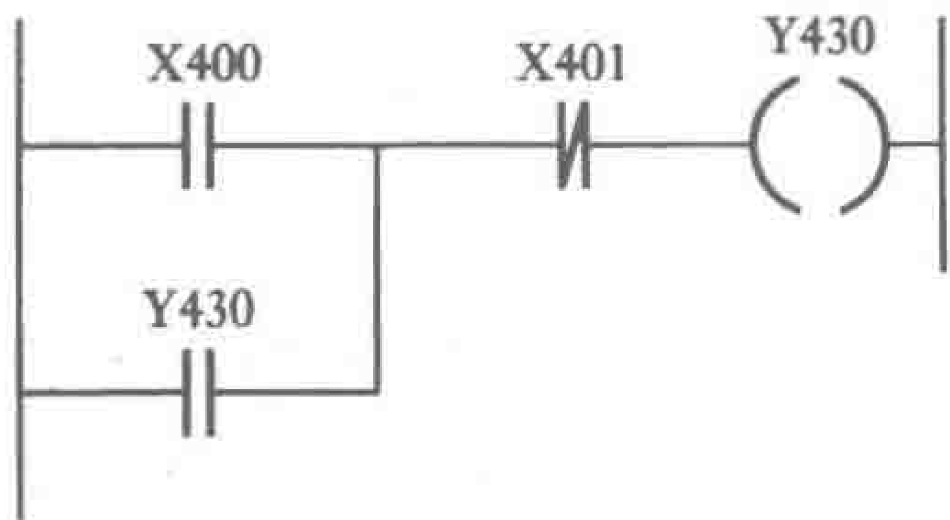


图 14-37　习题 14.3

14.4　请编写图 14-38 所示程序的指令，描述输出是如何随着时间变化的。

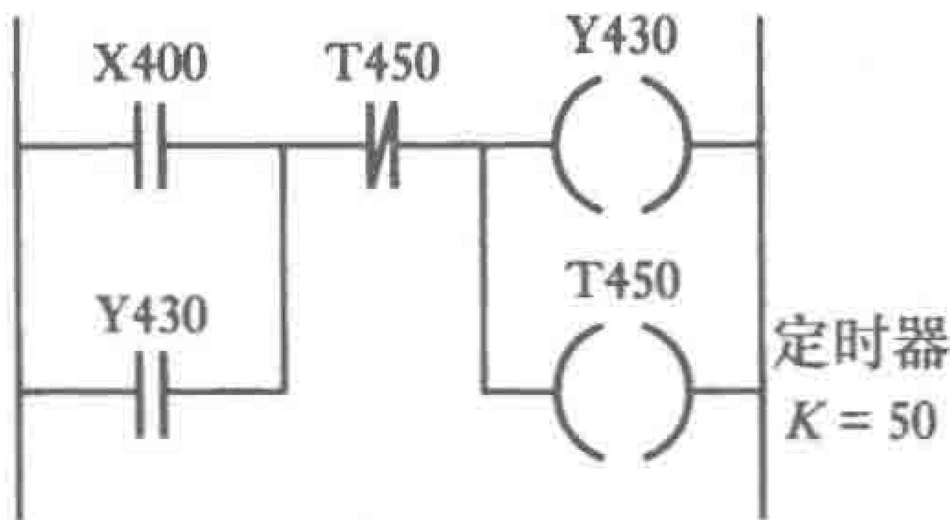


图 14-38　习题 14.4

14.5　请编写图 14-39 所示程序对应的指令，描述输入到 PLC 的结果。

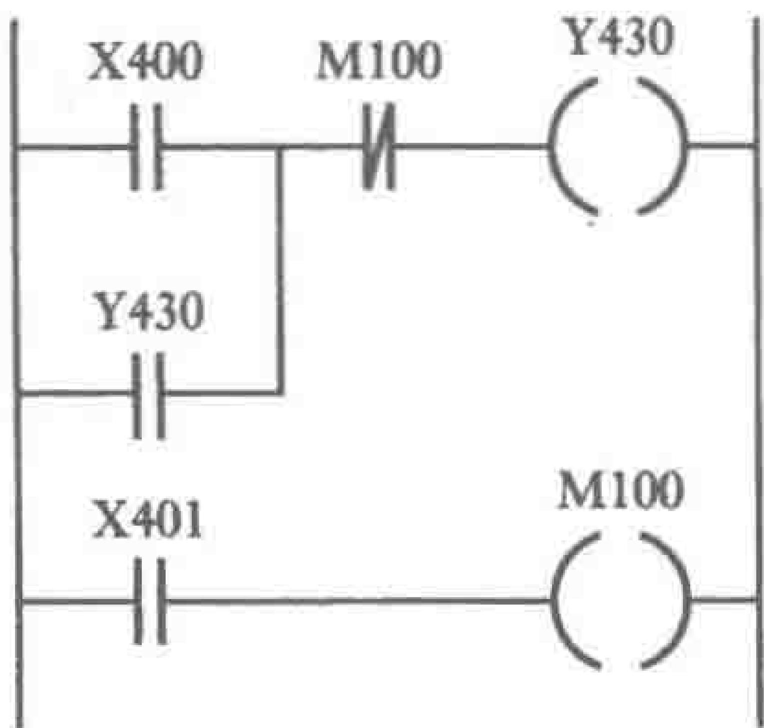


图 14-39　习题 14.5

14.6　请设计一个定时电路，该电路的输出切换为：打开 1s，断开 20s，再打开 1s，再断开 20s……

374

14.7　请设计一个定时电路，输出 10s 后，将之断开。

14.8　请设计一个电路，可以使电动机启动，经过 100s 延迟后，泵启动；当电动机关闭 10s 后，泵也关闭。

14.9　请设计一个电路，可用于家用洗衣机中，启动泵，在 100s 时间里向洗衣机中注入水，然后关闭并启动加热器，加热水 50s 后关闭。启动另一个泵，在 100s 的时间里将水抽空。

14.10　请设计一个电路，可用于将物体用传送带移动至工作站。产品出现在工作站是通过遮断一束光与光电传感器之间联系的方法来检测。传送带停止运行 100s 以便对产品进行操作。用常开开关来启动电动机，常闭开关来停止电动机。

14.11　数据输入 X400，如图 14-40 所示。请问图 14-27 中移位寄存器的时序图如何变化？

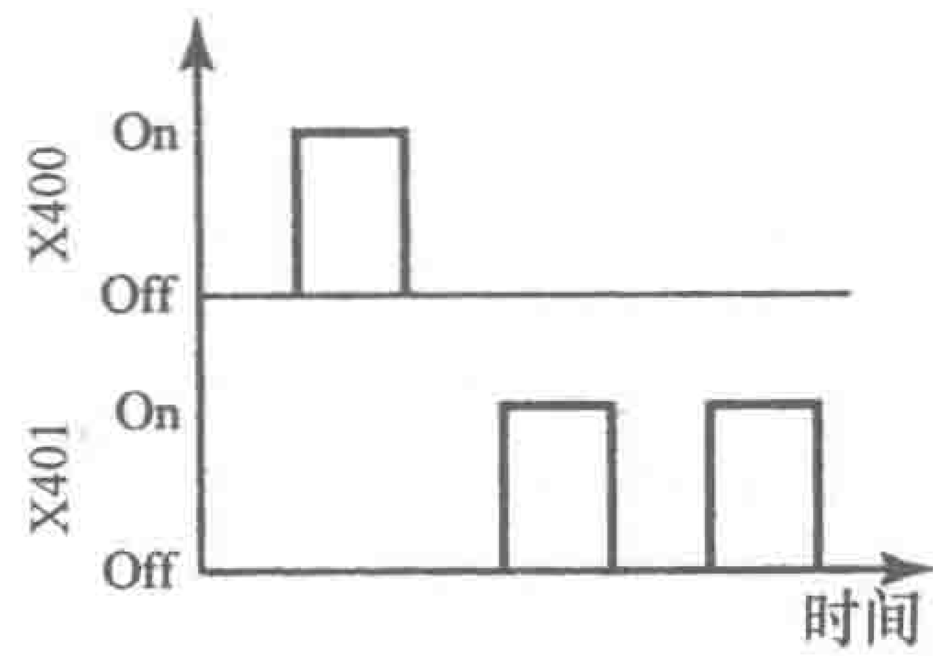


图 14-40 习题 14.11

14.12 请解释 PLC 如何处理模拟输入。

14.13 请设计一个可以用于控制汽缸中活塞运动的 PLC 系统，当瞬间按下第一个开关时，汽缸内活塞向一个方向移动，瞬间按下第二个开关时，活塞向另一个方向移动。提示：可用 4/2 螺线管控制阀。

14.14 请设计一个 PLC 系统，来控制使用 4/2 先导电磁阀的缸体内的活塞移动。当一个接近传感器在冲程的一端闭合触点时，活塞朝一个方向移动，而在冲程的另一端的接近传感器指示活塞到达时，向另一个方向移动。

375

第15章 通信系统

本章目的

通过本章的学习，读者将能够：

- 描述集中式、分层式与分布式控制系统，传输数据与协议所使用的网络配置与方法。
- 描述开放系统互联通信模型。
- 描述常用的通信接口：RS-232、IEEE 488、20mA 电流环、I^2C 总线和 CAN 总线。

15.1 数字通信

一条外部总线是一组信号线，其作用是将微处理器、微控制器、计算机与可编程逻辑控制器（PLC）互联起来，并同时将它们与外围设备连接起来。因此，如果想把一台计算机的输出结果传输至打印机并打印出来，需要一条总线将计算机连接到打印机。多处理器系统非常普遍。举例来说，在一辆轿车中可能有多个微处理器，每一个控制着系统中不同的部分，如发动机管理、制动与仪表盘，各个微处理器之间的通信是必要的。在自动化的工厂中，不仅需要将数据在可编程逻辑控制器、显示器、传感器与驱动器之间传输，允许操作人员输入数据与程序，而且还需要能够与其他计算机进行数据通信。例如，有可能需要将一台 PLC 连接到一个涉及多台 PLC 与计算机的控制系统中。计算机集成制造（CIM）就是一个涉及大量机器互联的大型网络的例子。无论是两台机器的简单连接，还是涉及大量机器互联的大型网络，本章将阐述如何实现这种计算机之间的数据通信，以及标准通信接口的构成。

15.2 集中式、分层式与分布式控制

集中式计算机控制（centralized computer control）利用一台中央计算机控制整个工厂。这种方式存在的问题是中央计算机的故障将会导致整个工厂失去控制。使用双计算机系统能够避免这种情况的发生。如果一台计算机发生故障，另一台能够接替它的工作。这种集中式系统在 20 世纪六七十年代十分普遍。微处理器的发展与不断降低的计算机价格使得多计算机系统变得更加普遍，并推动了分层式与分布式系统的发展。

376

在**分层式系统**（hierarchical）中，针对所完成任务的不同，各个计算机处于不同的层次中。主要用于完成例行任务的计算机受到主要负责制定决策的计算机的管理。例如，用于系统直接数字控制的计算机会受到对整个系统执行监督控制的计算机的管理。根据所涉及功能的不同，将工作分配给各个计算机。计算机变得专门化，有些计算机仅接收某些信息，而其他计算机仅接收另一些信息。

在**分布式系统**（distributed）中，每个计算机系统都完成与其他计算机系统本质上相似的任务。如果一台计算机发生故障或者超载，其工作可以转移给其他计算机。整个工作分布在所有计算机上，而不是根据所涉及的功能分配给特定的计算机。没有专门化的计算机，因此每台计算机都需要获取系统中所有的信息。

最现代化的系统通常是分布式系统与分层式系统的混合系统。例如，测量与驱动的工作可以分布到大量互联的微处理器/计算机上，它们能提供整个工厂的数据库。这些计算机可能被

一台用于直接数字控制或者顺序控制的计算机监督，而这台计算机本身又可能被一台用于监督控制整个工厂的计算机所管理。这种方案的典型层次结构如下：

第一层 测量与驱动

第二层 直接数字控制与顺序控制

第三层 监督控制

第四层 管理控制与设计

分布式/分层式系统具有允许控制系统中的测量扫描与信号调理任务被大量微处理器分担的优点。这涉及大量具有高扫描频率的信号。如果需要额外的测量环路，可以很简单地通过增加微处理器来增加系统的容量。测量单元可以很分散地布置在各个测量源附近。其中一个单元的故障不会导致整个系统的故障。

15.2.1 并行与串行数据传输

数据通信可以由并行或串行传输链路实现。

1. 并行数据传输

在计算机中，数据通常由**并行数据路径**进行传输。并行数据总线同时传输8、16或32位数据，每个数据位与控制信号有各自的总线导线。因此，如果需要传输8位数据，例如11000111，那么就需要8条数据导线。整个8位数据的传输时间与传输1位数据的时间是一样的，因为每一位都在一条并列的导线上。同时，还需要握手（参见13.3.2节）线路，每个通过线路传输的字符需要握手信号来指示该数据是可以被发送的，以及接收端已经准备好接收。377并行数据传输能够实现高数据传输率，但是所需要的电缆与电路使其变得昂贵。因此，通常用于短距离传输或必须要求高传输率的场合。

2. 串行数据传输

对于这种方式，数据与控制信号是以逐位顺序发送的方式在单条线路中传输。只需要两根导线，一根用于发送数据，另一根用于接收数据。由于一个数据字的所有位是按顺序的、而非同时的方式传输的，所以与并行数据传输相比，数据传输率要慢很多。然而，由于所需导线数量的大大减少，所以串行数据传输所需的费用更低。例如，在一辆轿车中使用了大量的微控制器，它们之间就是以串行数据传输的方式连接的。若不采用串行数据传输方式，将需要相当多的导线。串行数据传输通常用于除了最短距离外设连接之外的所有场合。

对于通过串行链路发送一系列字符的问题，接收端需要知道一个字符的起始与结束位置。串行数据传输可以是异步的或同步的。**异步传输**表明发送端与接收端计算机不同步，各自有自己独立的时钟信号。所传输字符之间相隔的时间是任意的。因此，沿着该链路传输的每个字符会在最前端附加一个起始位以告知接收端字符的起始位置，并在最末端附加一个停止位以表示结束位置。这种方法有一个缺点，即传输每个字符需要传输额外的位，因此，降低了线路数据传输的效率。而**同步传输**就不需要起始与停止位，因为发送端与接收端有共同的时钟信号，在一个周期中字符的传输始终会在相同的时刻自动开始与停止。

数据传输率以位每秒的方式来描述。如果传输一个单一符号的 n 位数据所用的传输时间是 T 秒，那么数据传输率就是 n/T。数据传输率的单位是**波特**。当传输的每个字符仅由一个符号表示时，波特率才与每秒传输的位数相同。因此，一个不使用起始与停止脉冲的系统的波特率与位速率相同，但是当存在起始与停止位时，这两者就不相同。

15.2.2 串行数据通信方式

串行数据传输有以下三种方式：

1. 单工方式

只能单向传输，从设备A到设备B，而设备B无法反向传输到设备A（见图15-1a）。你可

以把这种设备间的连接方式看作一条单行道。这种方式通常仅用于将数据传输到打印机等不会发送信息的设备。

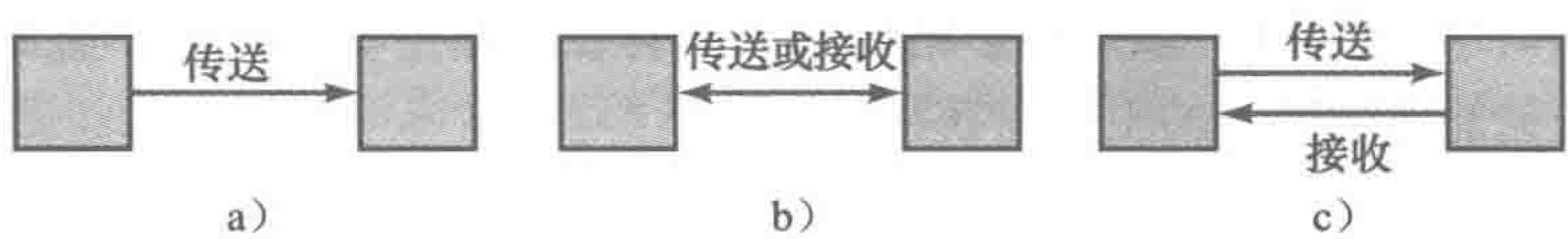


图 15-1 通信方式

378

2. 半双工方式

在某一时刻只能向一个方向传输数据，但是传输的方向是可以改变的（见图 15-1b）。链路两端的终端能够在发送与接收之间切换。因此，既能从设备 A 传输到设备 B，也能从设备 B 传输到设备 A，但是这两者不能同时进行。你可以把这种方式看作一条车道正在维修的两车道道路，通过交通管制使两个方向的车辆交替通行。民用波段（Citizens Band，CB）无线电对讲机就是半双工方式的一个例子，使用者可以收听或者说话，但是两者不能同时进行。

3. 全双工方式

数据可以同时在设备 A 与设备 B 之间进行双向传输（见图 15-1c）。这种方式就像一条两车道道路，两个方向的车辆都可以同时通行。电话就是一个全双工方式的例子，使用者可以同时收听与说话。

15.3 网络

网络是指一个允许两个或两个以上互联的计算机/微处理器进行数据交换的系统。链路连接的逻辑结构称为网络拓扑。我们用节点这个术语来描述网络中的一个点，有一条或多条通信线路终结于该点，或者一个单元在该点接入通信线路。常用的结构有如下几种：

1. 数据总线结构

这种结构有一条线性总线（见图 15-2a），所有的工作站都接入该总线。该系统通常用于多点终端群集。当节点间的距离大于 100m 时，这通常是优先选择的结构。

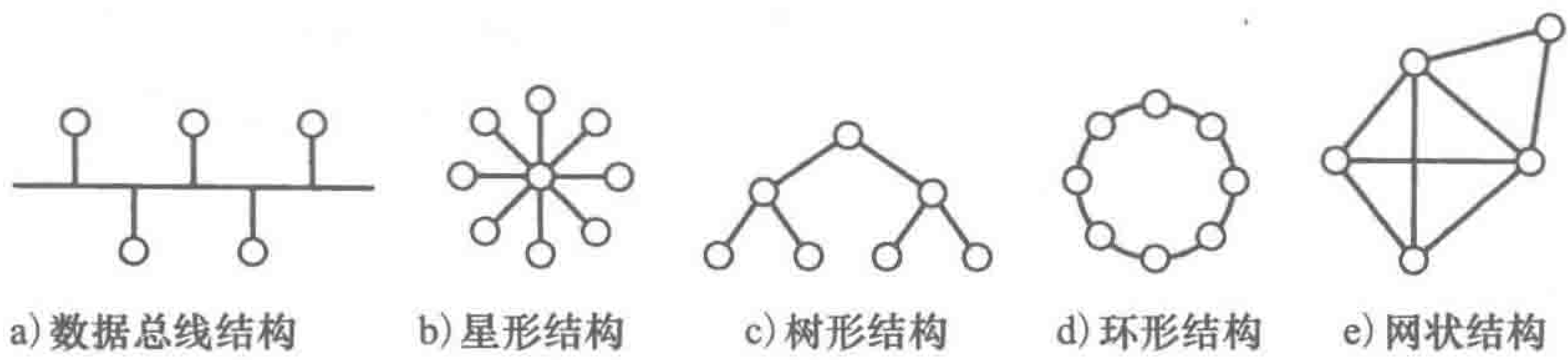


图 15-2 网络拓扑

2. 星形结构

这种结构的每个工作站之间都有专用的信道，并且所有通信都必须经过一个中央交换式集线器（见图 15-2b）。许多公司的电话系统（专用交换机（PBX））就是星形结构，所有的线路都要经过一个中央交换机。这种结构还通常用于将远程与本地终端连接到一个中央大型计算机上。这种结构的最大问题是一旦中央集线器发生故障，整个系统就趋于瘫痪。

3. 分层或树形结构

这种结构由一系列分支组成，它们间接汇聚到树形结构头部的一点（见图 15-2c）。任意两台工作站之间只有一条传输通路。这种结构可能由大量数据总线系统连接而成。与数据总线结构相似，这种结构也通常用于节点间的距离大于 100m 的场合。

379

4. 环形结构

这是一种非常流行的局域网拓扑结构，每台工作站都连接到一个环路上（见图15-2d）。节点间的距离通常小于100m。输入环形结构的数据会在环路中不断循环，直到被某些系统去除。环路中的数据能够被所有的工作站获得。

5. 网状结构

这种结构（见图15-2e）的工作站之间没有固定的连接方式，并且各个工作站之间会有多条数据通路。

我们用**局域网**（LAN）这个术语来描述一个覆盖局部地理区域（如某处的一栋或几栋建筑物）的网络。局域网的拓扑结构通常是总线、星形或环形结构。**广域网**（WAN）是一种覆盖一个国家乃至全世界的用于互联计算机、终端与局域网的网络。本章将主要讨论局域网。

15.3.1 网络访问控制

访问控制方式对于一个网络来说是很必要的，用于保证任何时刻只有一个用户能够发送数据。下面是常用的方式。

对于基于环形结构的局域网，主要有以下两种方式：

1. 令牌传递方式

这种方式使一个令牌（一个特殊的位序列）在环路中循环。当一台工作站想发送数据时，必须等待直到获得令牌，然后将令牌附加在数据的末端并发送出去。另一台想发送数据的工作站将令牌从数据包里去除，再将令牌附加在自己数据的末端并发送出去。

2. 时间片传递方式

这种方式使一些空的时间片在环路中循环。当一台工作站想传递数据时，将数据放入第一个到来的空时间片中，并发送出去。

对于总线或树形拓扑结构的网络，通常使用以下方式：

3. 带冲突检测的载波侦听多路访问（CSMA/CD）

这种方式通常用于**以太网局域网总线**。该方式要求工作站在发送数据前必须侦听其他的传输，任何一台工作站都能控制网络并发送数据，即多路访问。如果没有侦听到任何活动，则可以发送数据。如果侦听到活动，那么该工作站将等待直到没有侦听到任何活动。尽管在发送前先侦听，但仍然有可能出现两台或多台工作站同时开始发送的情况。如果侦听到这种情况，所有发生冲突的工作站都将立即停止发送并等待一个随机时间再尝试重新发送。

15.3.2 宽带与基带

380

宽带传输是指在一个网络中将信息调制到一个射频载波上，并能通过同轴电缆等介质进行传输。宽带局域网的典型拓扑结构是带有分支的总线拓扑结构。宽带传输允许同时传输大量被调制的射频载波，因此具有多信道通信能力。**基带传输**是指直接通过通信介质传输数字信息。基带传输网络任意时刻只支持传输一个信息信号。一个局域网既可以是宽带传输，也可以是基带传输。

15.4 协议

被传输的数据将包含两种类型的信息。一种是一台计算机希望发送给另一台的数据，另一种是称为**协议数据**的信息，计算机与网络之间的接口使用协议数据来控制计算机与网络之间的数据传输。协议是一组正式规则，用于管理数据格式、定时、排序、访问控制与错误控制。协议包括以下三个元素：

1）语法，定义数据格式、编码与信号电平。

2）语义，负责同步、控制与错误处理。

3）定时，负责数据的排序以及数据率的选择。

当一个发送端与一个接收端通信时，它们必须采用同样的协议，例如两个以串行方式传输数据的微控制器。在单工方式中，数据块只能从发送端传输到接收端。然而，在半双工方式中，每个传输的数据块，如果是有效的，必须被接收端确认（ACK）才能进行下一个数据块的发送（见图 15-3a）；如果是无效的，接收端会发送一个否定的确认信号（NAK）。因此，无法实现数据的连续传输。在一个数据块之后紧接着发送**循环冗余校验**（CRC）位，是一种错误检测方式。数据是以二进制数字形式进行传输，在发送端数据除以一个数后所得的余数作为循环校验码。接收端将所收到的包含 CRC 码的数据除以同一个数，如果余数是 0，则说明信号没有错误。在全双工方式中（见图 15-3b），数据可以同时发送与接收。

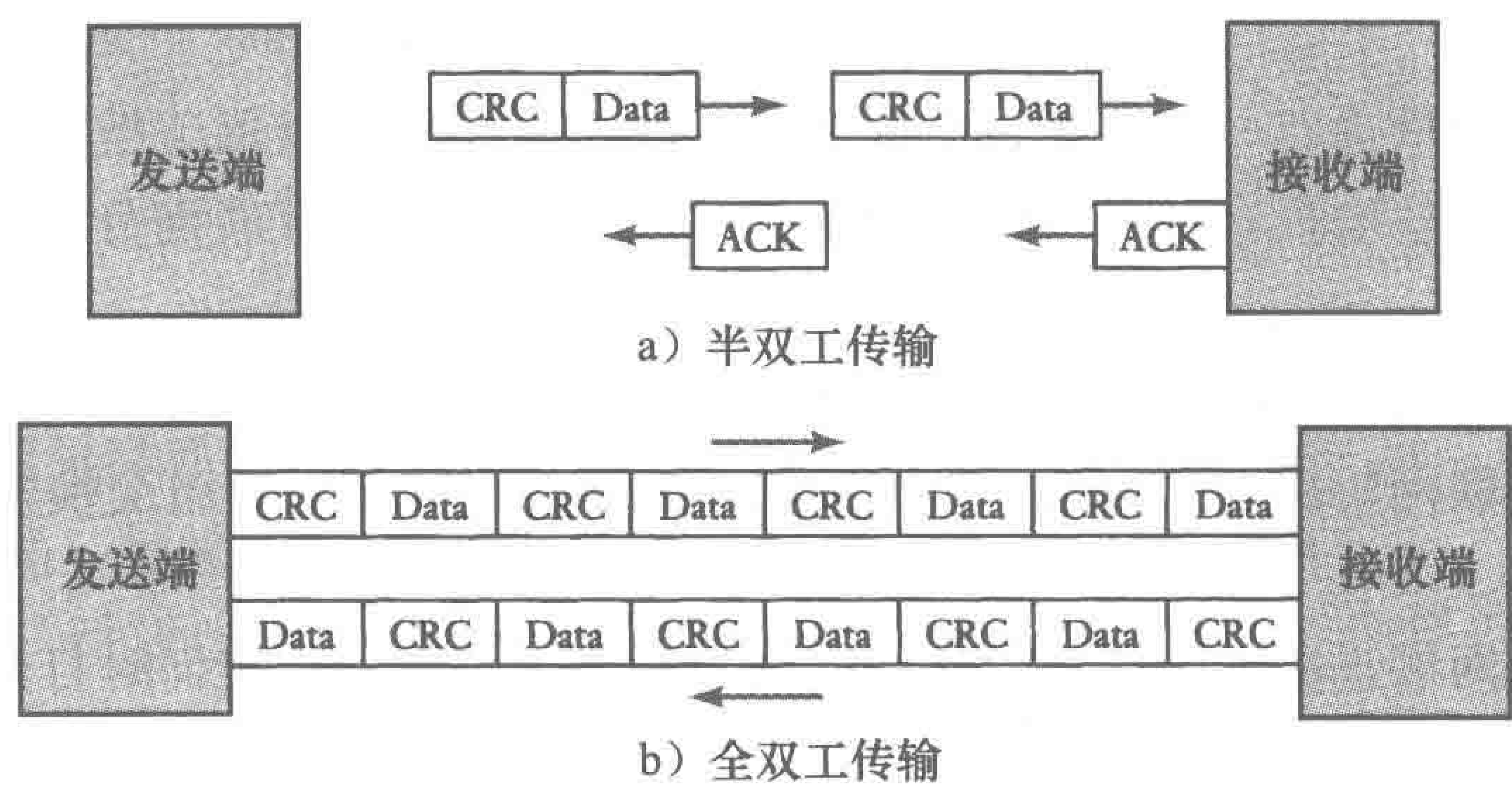


图 15-3　协议

381

在一个发送的数据包中，需要包含协议信息。例如，在异步传输方式中，可能需要用于指示数据开始与结束的字符。对于同步传输与**二进制同步通信协议**（Bisync Protocol），一个数据块之前有同步序列位，通常是 ASCII 字符 SYN（见图 15-4a）。接收端使用这些 SYN 字符来实现字符同步，准备接收 8 位一组的数据。摩托罗拉 MC6852 是一个同步串行数据适配器（SSDA），该适配器用 6800 微处理器以提供使用二进制同步通信协议的同步串行通信接口。它与在 13.5 节中所描述的异步通信接口适配器相似。另一个协议是**高级数据链路控制协议**（HDLC）（见图 15-4b）。这是一个全双工协议，每条报文的开始与结尾由位序列 01111110 来指示。在开始标志之后是地址与控制字段。地址字段指明目的工作站的地址，控制字段定义该帧是监控帧、信息帧还是无序号帧。在报文之后是一个用于给出 CRC 码的 16 位帧确认序列。摩托罗拉 6854 就是一个使用 HDLC 协议的串行接口适配器的例子。

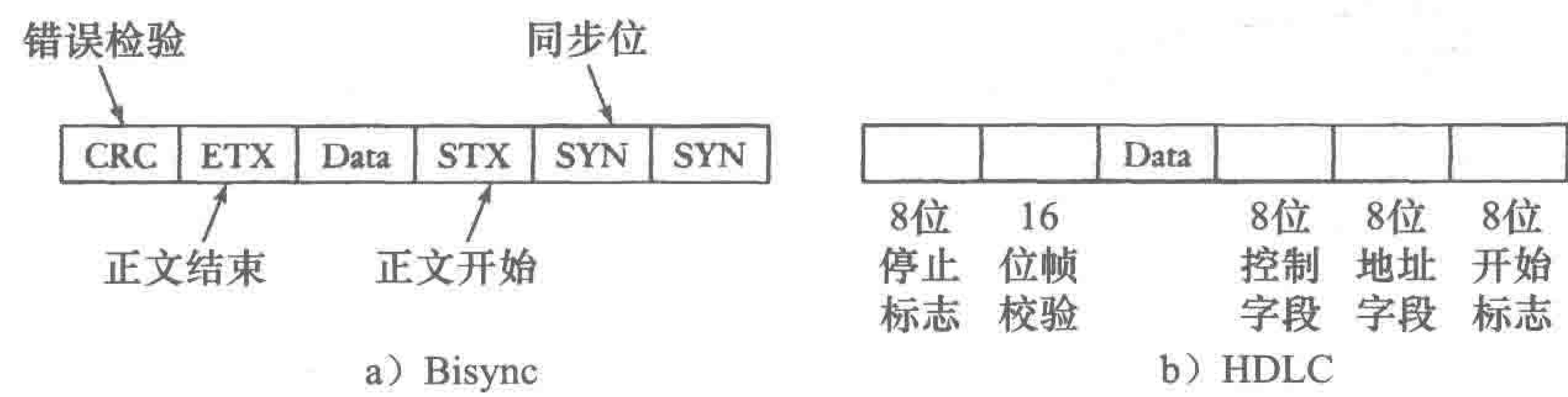


图 15-4　两种协议

15.5　开放系统互连通信模型

通信协议需要存在于多个层次之上。国际标准化组织（ISO）定义了一个 7 层标准协议系统，也就是**开放系统互连**（OSI）模型。这个模型是开发共同协调标准系统的框架。模型的各

层概述如下：

1. 物理层

该层描述了网络中各物理组件之间的位传输方式。它主要负责硬件问题，例如电缆与接头的种类、同步数据传输与信号电平等。常用的定义于物理层的局域网系统包括以太网与令牌环网。

2. 数据链路层

该层定义了用于发送与接收报文、检测与修正错误、对所发送数据进行顺序控制的协议。该层主要负责将数据打包放入电缆中，以及在接收端将数据包从电缆中取走。以太网和令牌环网也定义于该层。

382

3. 网络层

该层负责处理通信通路与网络中报文的寻址、路由与控制，以保证报文能被传送到正确的目的地。常用的网络层协议有互联网协议（IP）与 Novell 公司的互联网分组交换协议（IPX）。

4. 传输层

该层提供可靠的端到端报文传输。该层负责建立与保持发送端与接收端之间的连接。常用的传输层协议是互联网传输控制协议（TCP）与 Novell 公司的序列分组交换协议（SPX）。

5. 会话层

该层负责在由网络连接的应用进程之间建立对话。该层负责决定何时打开或关闭两台工作站之间的通信。

6. 表示层

该层负责使所传输的编码数据能够以适合用户操作的方式表示。

7. 应用层

该层提供实际用户信息处理功能与应用程序相关的服务。该层提供如文件传输或者电子邮件等功能，一台工作站利用这些功能与网络中其他工作站进行通信。

15.5.1 网络标准

基于 OSI 层模型的常用网络标准有很多种，以下给出了一些例子。

在美国，通用汽车公司（GM）曾意识到它的生产活动存在设备使用各种非标准协议的问题。通用汽车公司为此开发了一个用于工厂自动化的标准通信系统。该标准称为**制造自动化协议**（MAP）（见图 15-5）。为了使系统能够适应生产环境，各层协议的选择需要满足相应的要求。第 1 层和第 2 层以电子硬件的方式实现，第 3 ~ 7 层以软件方式实现。在物理层中使用的是宽带传输方式。宽带传输方式使系统能够用于多种服务（包括 MAP 通信所需的服务）。在数据链路层采用令牌总线方式，并使用逻辑链路控制（LLC）来实现错误检测等功能。其他层采用 ISO 标准。在第 7 层，MAP 引入了制造报文服务（MMS），这是一个与工厂车间通信有关的应用程序，它定义了可编程逻辑控制器与数控机器或机器人之间的相互交互。

383

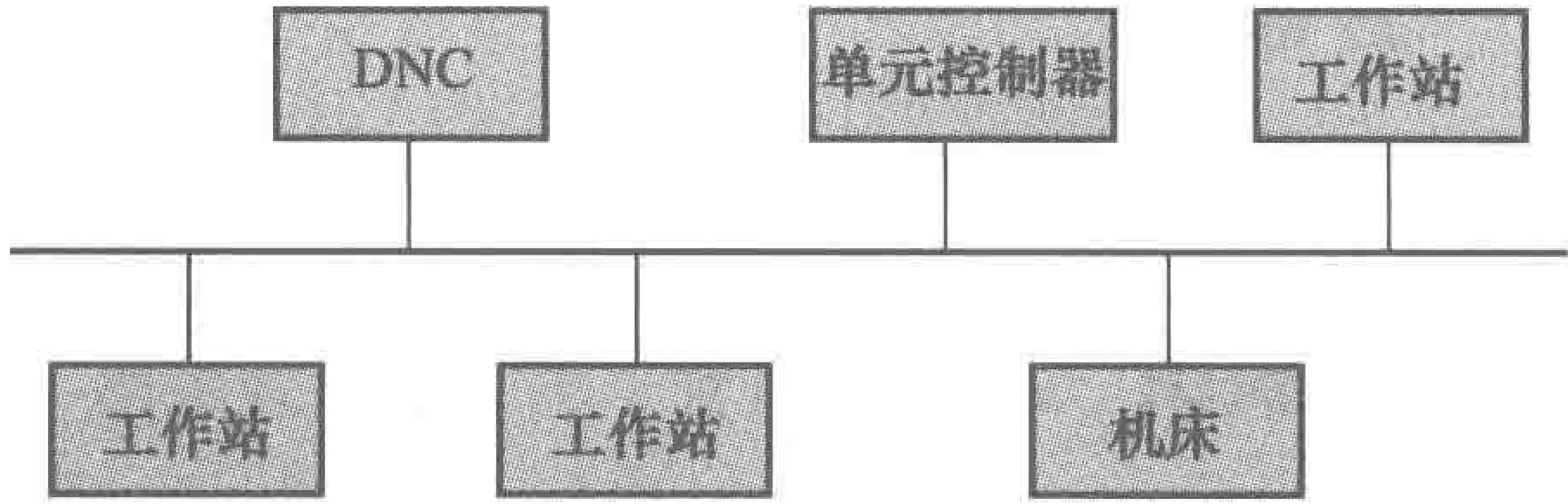


图 15-5 MAP

技术和办公系统协议（TOP）是由波音计算机服务公司开发的一种网络标准。它与 MAP 有许多共同之处，但能以较低的费用实现，因为它是一个基带传输系统。它在第 1～2 层与 MAP 不同，采用令牌环网或 CSMA/CD 总线。在第 7 层，它制定了用于办公室而不是工厂车间的应用程序协议。由于采用了 CSMA/CD 方式，工作站在发送数据前需要侦听其他传输活动。TOP 与 MAP 网络是兼容的，一个网关设备能够把 TOP 与 MAP 网络连接起来。该设备完成相应的地址转换与协议改变。

系统网络体系结构（SNA）是由 IBM 开发的一个作为 IBM 产品设计标准的系统。SNA 划分为 7 层，然而却与 OSI 的各层有所不同（见图 15-6）。数据链路控制层为令牌环局域网提供支持。SNA 的 5 个层合并为两个单元：第 2～3 层作为通路控制网络，第 4～6 层作为网络可寻址单元。

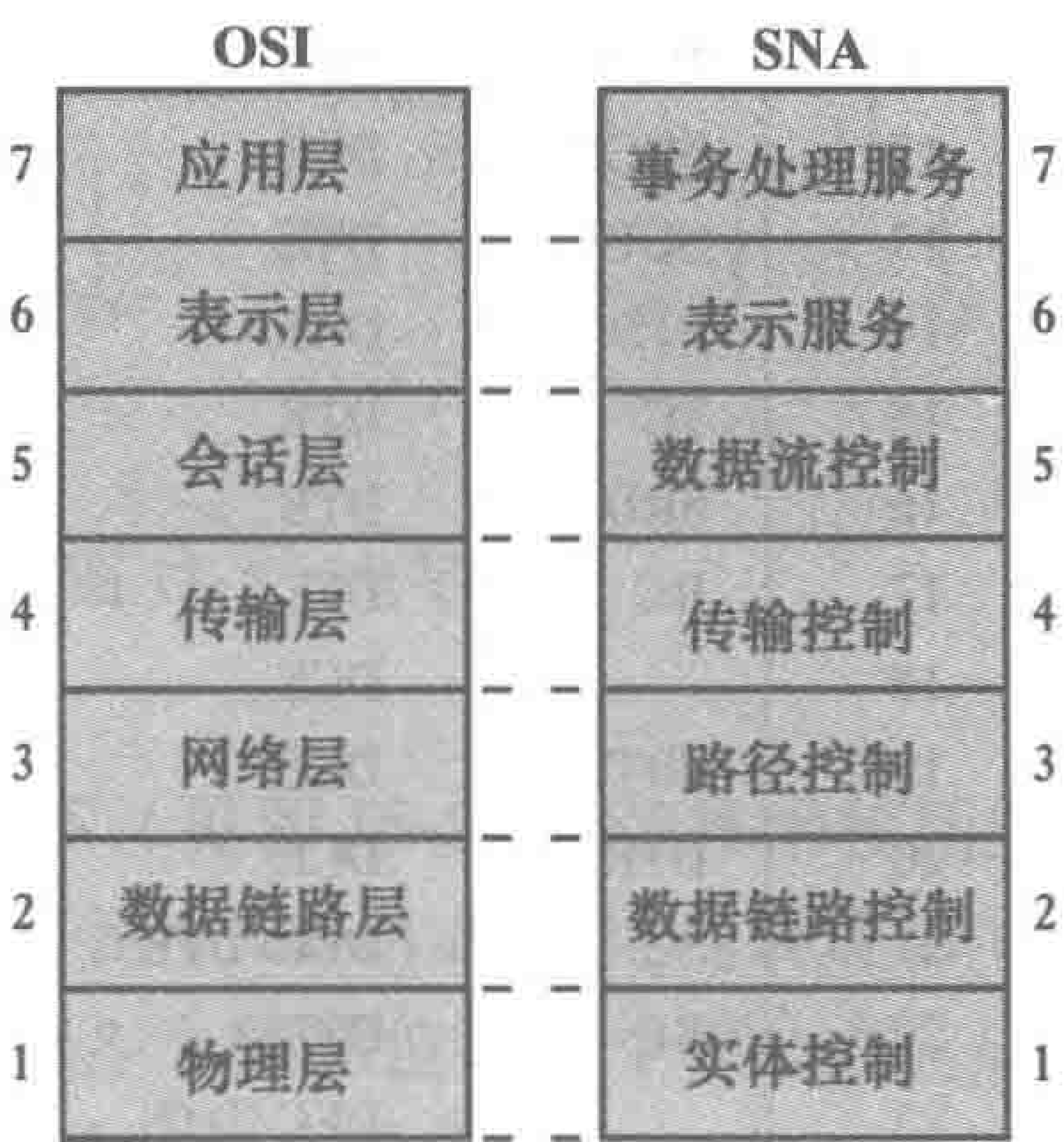


图 15-6 SNA

在 PLC 系统中，由 PLC 制造商销售的网络通信系统是相当普遍的。例如，Allen-Bradley 公司的 **Allen-Bradley 数据高速公路**，使用令牌传递方式来控制报文发送；三菱公司的 Melsec-Net，与德州仪器公司的 TIWAY。PLC 网络常用的系统是以太网。这是一种采用 CSMA/CD 访问控制方式的单总线系统，该系统广泛用于涉及 PLC 与计算机通信的系统。尽管当网络流量低时，采用 CSMA/CD 方式能够很好地工作。但采用该方式存在的问题是，随着网络负载的增大，冲突的次数与相应的发送端后退的次数随之增加，因此网络吞吐量会大大降低。

384

15.6 串行通信接口

串行接口包括同步与异步协议。常用的异步接口有 RS-232 及其更新版本、20mA 电流环、I^2C、CAN 和 USB。

15.6.1 RS-232

最常见的串行接口是 RS-232，该接口最早由美国电子工业协会（EIA）于 1962 年定义。该标准与数据终端设备（DTE）和数据电路端接设备（DCE）有关。数据终端设备能够通过接口发送或接收数据，例如微控制器。数据电路端接设备用于通信，典型例子是调制解调器。它是微型计算机与传统模拟电话线路之间进行连接所必不可少的组成部分。

RS-232 信号可以分为以下三类：

1. 数据信号

RS-232 提供两组独立的串行数据通道，分别是主通道与辅助通道。这两个通道都能以全双工方式工作。

2. 握手控制信号

握手信号用于控制通信路径中的串行数据流。

3. 定时信号

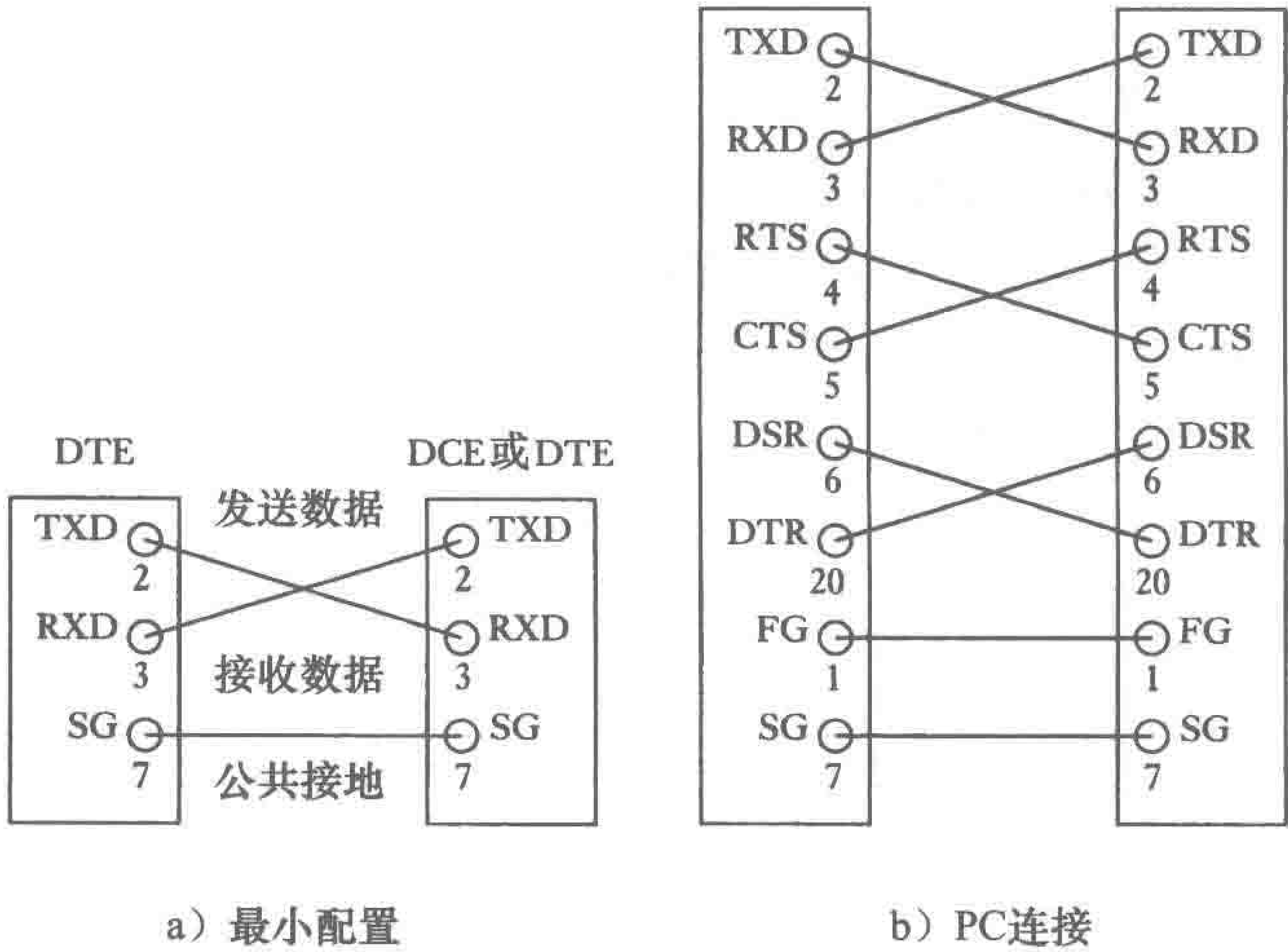
对于同步传输方式，需要在发送端与接收端之间传递时钟信号。

表 15-1 给出了 RS-232C 连接器的引脚编号与定义。在一个特定的连接中，并不需要用到所有的引脚与信号。信号地导线提供返回路径。RS-232C 串口采用 25 针 D 型连接器，通常在电缆上使用公插头，而在 DCE 或 DTE 上使用母插座。

表 15-1 RS-232 引脚定义

引脚	简写	方向：至	信号/功能
1	FG		Frame ground 屏蔽地
2	TXD	DCE	Transmitted data 发送数据
3	RXD	DTE	Received data 接收数据
4	RTS	DCE	Request to send 请求发送
5	CTS	DTE	Clear to send 清除发送
6	DSR	DTE	DCE ready DCE 准备好
7	SG		Signal ground/common return 信号地/公共返回
8	DCD	DTE	Received line detector 数据载波检测
12	SDCD	DTE	Secondary received line signal detector 数据载波检测（辅助）
13	SCTS	DTE	Secondary clear to send 清除发送（辅助）
14	STD	DCE	Secondary transmitted data 发送数据（辅助）
15	TC	DTE	Transmit signal timing 发送信号时钟
16	SRD	DTE	Secondary received data 接收数据（辅助）
17	RC	DTE	Received signal timing 接收信号时钟
18		DCE	Local loop-back 本地环路控制
19	SRTS	DCE	Secondary request to send 发送请求（辅助）
20	DTR	DCE	Data terminal ready 数据终端准备好
21	SQ	DEC/DTE	Remote loop-back/signal quality detector 远端环回/信号质量检测器
22	RI	DTE	Ring indicator 振铃指示器
23		DEC/DTE	Data signal rate selector 数据信号率选择器
24	TC	DCE	Transmit signal timing 发送信号时钟
25		DTE	Test mode 测试模式

对于最简单的双向链路，只需要两条导线连接引脚 2 与 3，用于发送与接收数据，以及一条连接信号地（引脚 7）的导线，用于信号返回（见图 15-7a）。因此，最简单的连接方式只需要一条三芯电缆。对于一个连接个人计算机（PC）与视频显示单元（VDU）的简单连接，需要用到引脚 1、2、3、4、5、6、7 与 20（见图 15-7b）。引脚 4、5、6、20 信号用于确认接收端是否准备好接收信号；发送端是否准备好发送信号；数据是否准备好被发送。



a）最小配置　　b）PC连接

图 15-7 RS-232 连接方式

需要注意 RS-232 接口的传输距离，因为当电缆长度大于 15m 时，噪声干扰限制了高位速率的传输。最高数据传输率大约是 20kbit/s。与 RS-232 类似的其他标准，例如 RS-422 与

RS-485，能够实现更高的传输速率与更长的传输距离。

RS-422 接口为每个信号使用一对线路，能够用于远至 1220m 的连接场合或在干扰更严重的环境中实现高达 100bit/s 的传输速率。然而，最快速率与最远距离不能同时实现。RS-485接口能够在远至 1220m 的连接场合中实现 100kbit/s 的传输速率。

385
~
386

摩托罗拉微控制器 MC68HC11 的串行通信接口能够在多种波特率下以全双工方式工作。然而，该系统的输入与输出使用晶体管－晶体管逻辑（TTL），逻辑 0 是 0V，逻辑 1 是 + 5V。RS-232C 标准的逻辑 0 是 + 12V，逻辑 1 是 - 12V。因此，需要对信号电平进行转换，这可由使用集成电路设备来实现。例如，MC1488 用于 TTL 到 RS-232C 的转换，MC1489 用于 RS-232C 到 TTL 的转换（见图 15-8）。

图 15-8 电平转换

15.6.2 20mA 电流环

另一项基于于 RS-232，但不属于该标准的技术是 20mA 电流环（见图 15-9）。该技术使用电流信号而非电压信号。一对分离的导线用于发送和接收数据，环路中 20mA 的电流代表逻辑 1，0mA 代表逻辑 0。串行数据按以下方式进行编码：一个起始位、8 个数据位与两个停止位。这种电流信号能够传输几千米的距离，大大超过标准 RS-232 电压连接的传输距离。

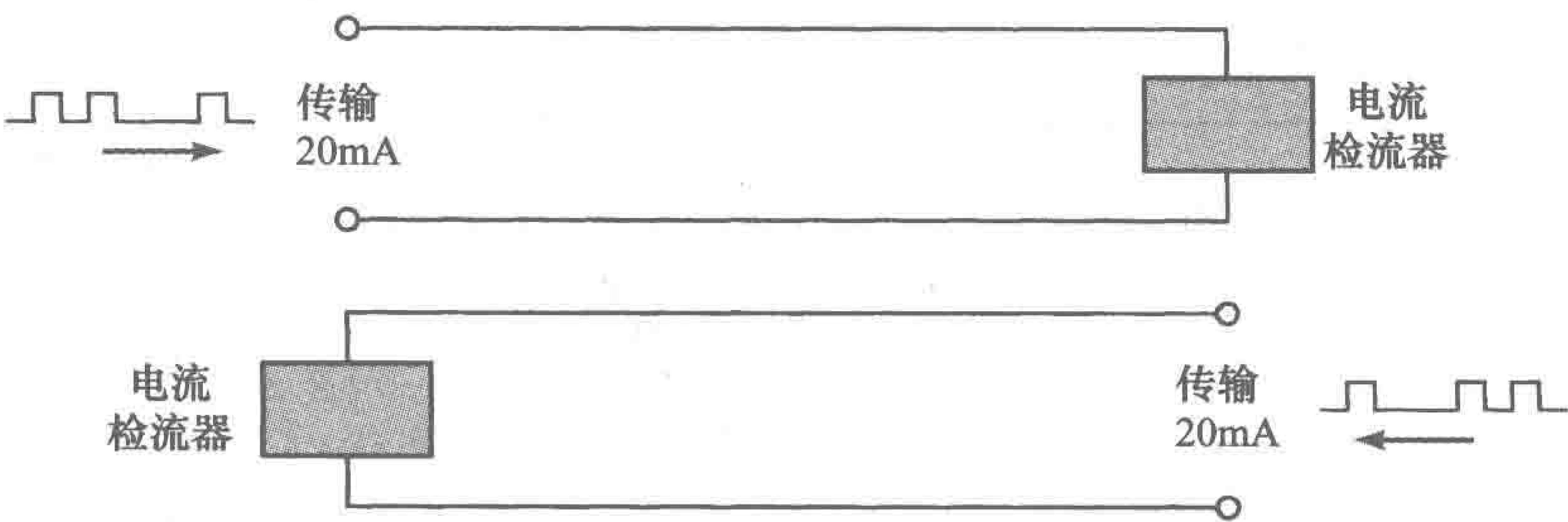


图 15-9 20mA 电流环路

15.6.3 I²C 总线

集成电路内部通信（Inter-IC Communication，I²C）总线，是一个由 Philips 公司开发的用于集成电路或模块间通信的串行数据总线。该总线只需要两条导线就能在设备间交换数据与指令。这使得电路得到很大程度的简化。

387

这两条线路分别是双向数据线（SDA）与时钟线（SCL），并都通过电阻连接到电源正极（见图 15-10）。产生报文的设备是发送端，接收报文的设备是接收端。控制总线操作的设备是主机，被控制的设备是从机。

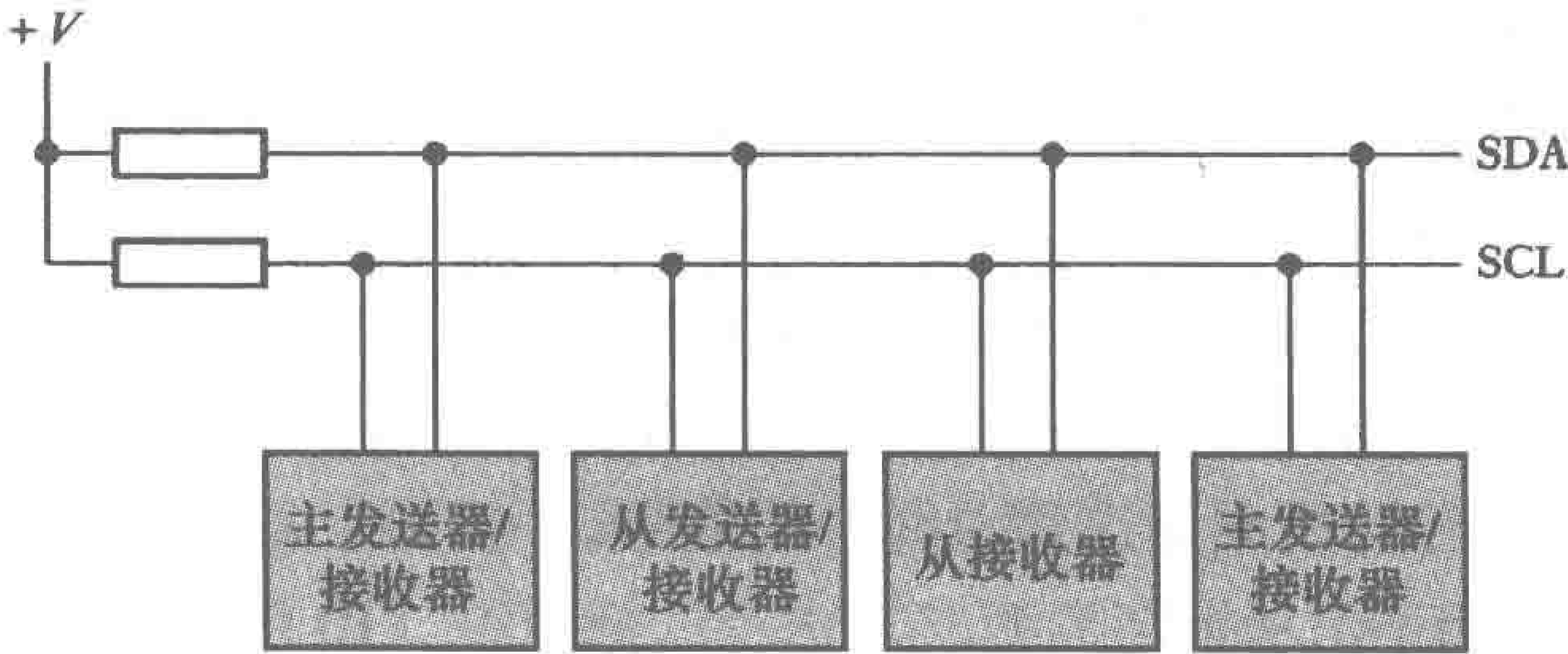


图 15-10 I²C 总线

使用的协议如下：数据传输只能在总线空闲时进行。在数据传输过程中，数据线必须保持稳定，而此时时钟线是高电平。当时钟线是高电平时，数据线的变化将认为是控制信号。

1）当数据线与时钟线都是高电平时，总线是空闲的。

2）当时钟线是高电平时，数据线从高电平变为低电平用于定义数据传输的起始条件。

3）当时钟线是高电平时，数据线从低电平变为高电平用于定义数据传输的停止条件。

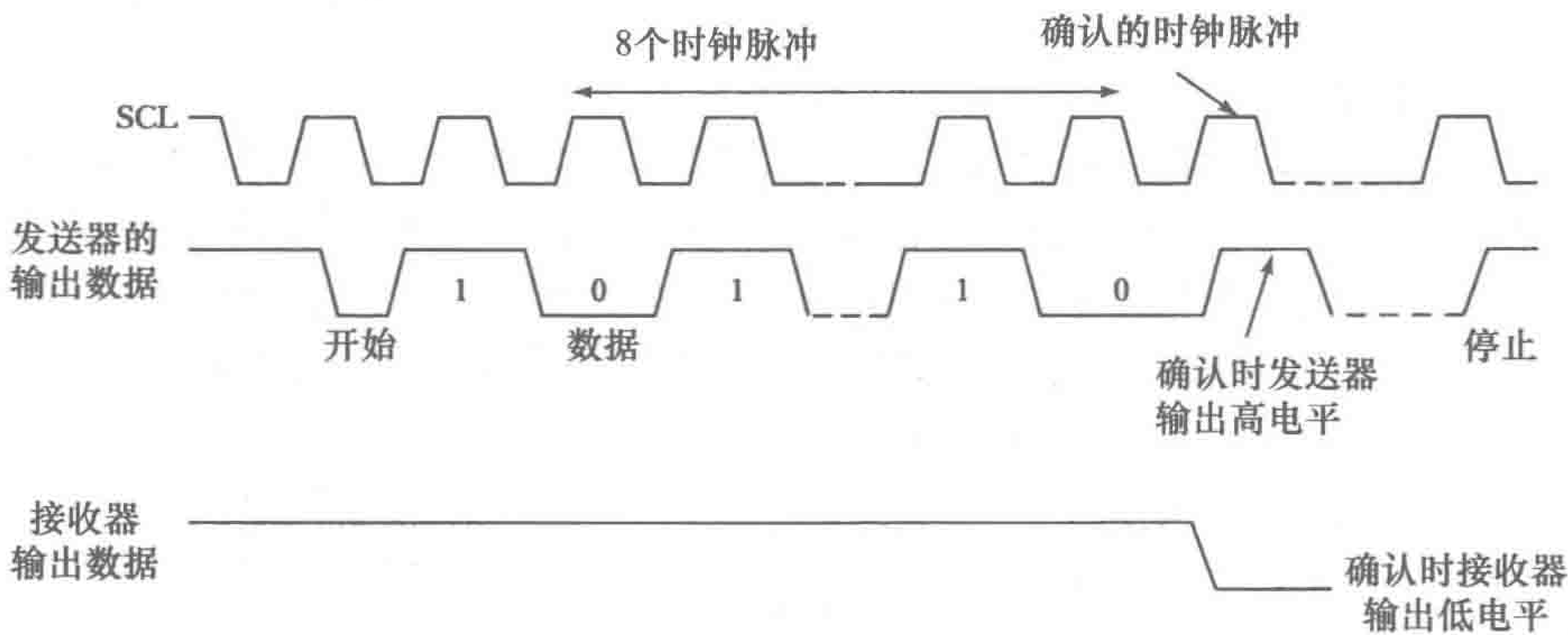
4）数据在起始与停止条件之间传输。

5）在起始条件之后，数据线在时钟信号的高电平周期内保持稳定，只有时钟信号的低电平周期内才能改变高低电平状态。

6）每传输一个数据位都有一个对应的时钟脉冲，在起始与停止条件之间所传输的字节数是没有限制的。接收端每接收一个字节的数据，就会以发送第9位的方式进行确认。

7）总线上的确认位对于发送端是高电平，对于接收端是低电平。

图15-11给出了上述协议所对应的时钟信号、发送端与接收端的输出波形。



388

图15-11　总线状态

15.6.4　CAN总线

一辆新型轿车上可能会有多达70个电子控制单元（ECU）用于控制各个子系统，例如发动机管理系统、防抱死制动、牵引力控制、主动悬架、安全气囊、巡航控制、车窗控制等。这可能会涉及大量的接线工作。然而，另一种解决方案是使用公共数据总线，使数据在总线中传输，轿车的各个部分都能获取总线中的数据。Bosch公司为此开发了**控制器局域网络**（CAN）协议。CAN总线现在也应用于其他自动化系统，作为现场总线。

CAN总线是一个用于连接ECU的多主机串行总线。该系统中的每个节点都能发送与接收报文，而且还需要以下设备：

1）一个主处理器，用于判断所接收的报文是什么意思，以及它想发送什么报文。传感器、驱动器与控制设备并不直接连接到CAN总线，而是连接到一个主处理器和一个CAN控制器。

2）一个CAN控制器，用于存储从总线串行接收的位，直到完成整个报文的接收。在CAN控制器触发一个中断请求之后，主处理器才能将报文取走。准备好以串行方式发送到总线中的报文也存储在该控制器中。

3）一个收发器，该收发器可能集成在CAN控制器中，用于将从总线接收到的信号电平转换为CAN控制器所使用的电平，并且它还有用于保护CAN控制器的保护电路。它还将来自CAN控制器的发送位信号转换成发送到总线中的信号。

每条报文都有一个用于识别报文类型或发送者的标识符（ID）字段，最多有8个数据字节。然而，如果两个或更多节点同时开始发送报文，那么就需要一些仲裁方法。一种非破坏性的仲裁

方法用来决定哪个节点能够发送。报文标识符字段中的0起决定作用，决定哪个节点赢得冲突并发送。因此，当一个节点将一位数据发送到总线中时，若检测到总线中已经有一个更高优先级的节点，那么该节点将停止发送并等待直到当前发送结束，然后再尝试发送自己的数据。例如，假设我们有两个报文，报文1的11位ID是11001100110，报文2的ID是10001101110，当传输到第4位时，仲裁方法判定报文1拥有更高优先级，因此报文2停止发送。

串行传输的标准CAN总线数据帧由夹在一个起始位、一个确认发送、帧结束位之间的报文组成。报文包括以下内容：

1）一个12位ID，最后一位是远程发送请求位。

2）一个6位控制字段，包括一个标识符扩展位、一个保留位和4位用于指明数据字节数的数据长度码。

3）数据字段。

4）一个16位CRC字段，即用于错误检测的循环冗余校验字段。

389

15.6.5 USB

通用串行总线（USB）用来使显示器、打印机、调制解调器和其他输入设备能够轻松地连接到PC，也就是即插即用。USB使用星形拓扑结构（参见15.3节），因此只需要一台设备插入PC中，其他设备能够插入到集线器中，从而形成层次化星形结构。因此，我们在PC上有一个主机集线器，其他外部集线器能够连接到主机集线器上。每个端口是一个四引脚插座，两个引脚用于供电，另两个引脚用于通信。USB 1.0和2.0为USB设备提供5V电源，但最大电流为500mA。若USB设备所需电源功率超过一个端口所能提供的，则可以使用外部电源供电。

低速版本USB1.0规范发布于1996年，传输速率为12Mbit/s，电缆长度最长为3m。高速版本USB2.0规范发布于2000年4月，传输速率为480Mbit/s，电缆长度最长为5m。由于最多支持5个USB集线器，因此一长串电缆与集线器可以覆盖远至30m的距离。超速USB 3.0规范由Intel及其合作公司于2008年8月发布，传输速率为4.8Gbit/s，使用该标准的产品现在已经在上市。USB 1.0与USB 2.0是半双工传输方式，USB 3.0可以实现全双工传输方式（参见15.2.2节）。

根集线器（root hub）拥有对所有USB端口的完全控制。它发起所有集线器与设备的通信。任何USB设备在向总线发送数据前必须向集线器控制器发送请求。在USB 2.0中，主控制器通过轮询总线了解流量。在USB 3.0中，连接的设备能够向主机请求服务。当一个USB设备第一次连接到一个USB主机时，主机开始一个枚举过程，并发送一个复位信号给USB设备。复位之后，主机读取该USB设备的信息，并分配给该设备一个唯一的7位地址。如果主机支持该设备，那么与该设备通信所需的驱动程序将被加载。驱动程序用于提供有关设备需求的信息，例如速度、优先级、设备功能以及数据传输时的数据包大小。

当应用程序的软件想从一台设备发送或接收一些信息时，它通过设备驱动程序发起一次传输。然后，驱动程序软件将该请求与其他设备驱动程序的请求一起放在内存中的某个位置。接下来，主控制器处理所有的请求，并将它以串行方式发送到主集线器端口。由于在USB总线上的所有设备都是并联的，所以所有设备都能接收到该信息。主机等待响应，有关设备以适当的信息回复主机。

发送的数据包有3种基本类型，即握手包、令牌包和数据包，各自有不同的格式与CRC（循环冗余校验，参见15.4节）。令牌包有4种类型：起始帧（SOF）包、IN与OUT包用于命令设备发送或接收数据、SETUP包用于设备初始化。

15.6.6　相线

火线是苹果公司开发的串行总线，该总线规范由 IEEE 1394 提供。它支持即插即用，用于磁盘驱动器、打印机与照相机等应用场合。

390

15.7　并行通信接口

连接打印机所用的并行接口通常是 Centronics 并行接口。然而，对于仪器，通信中最常用的并行接口是**通用仪器总线**（GPIB），IEEE 488 标准，最初由惠普公司开发用于连接计算机与仪器，因此通常也称为**惠普仪器总线**（Hewlett Packard Instrumentation Bus）。每台接入总线的设备称为“听者”（listener）、“讲者”（talker）或“控制者”（controller）。“听者”是从总线接收数据的设备，“讲者”根据请求向总线发送数据，“控制者”通过向“讲者”和“听者”发送命令的方式控制总线中的数据流，并采用轮询的方式确定哪些设备是活动的（见图 15-12a）。

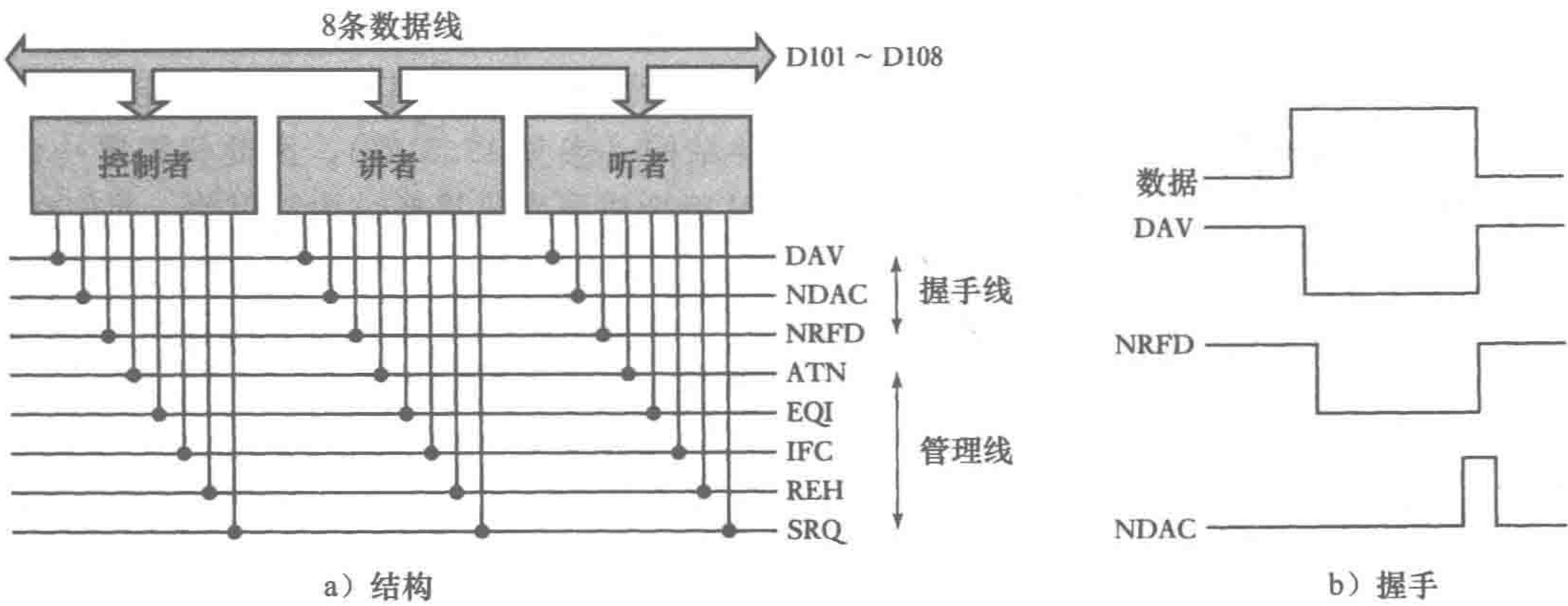


图 15-12　GPIB 总线

该接口共有 24 条线路：

1）8 条双向线路用于总线中不同设备之间传输数据与指令。

2）5 条线用于控制与状态信号。

3）3 条线用于设备间握手。

4）8 条线是接地返回线。

表 15-2 列出了各条线的功能以及在 25 针 D 型连接器中的引脚编号。最多可以有 15 台设备同时接入该总线，每台设备都有自己的地址。

表 15-2　IEEE 488 总线系统

引脚	信号组	简写	功能
1	数据	D101	数据线 1
2	数据	D102	数据线 2
3	数据	D103	数据线 3
4	数据	D104	数据线 4
5	管理	EOI	结尾或确定身份。“讲者”示意一个报文序列的结尾，或者控制器让一个设备表明身份
6	握手	DAV	数据有效。当该线为低电平时，数据线中的信息是有效的和可以被接收的
7	握手	NRFD	没有准备好接收数据。“听者”将该线置为高电平以示意已准备好接收数据

（续）

引脚	信号组	简写	功能
8	握手	NDAC	没有可以接收的数据。“听者”将该线置为高电平以表示数据已被接收
9	管理	IFC	接口清除。控制者将系统中的所有设备复位为初始状态
10	管理	SRQ	服务请求。设备向控制者表示其需要关注
11	管理	ATN	注意。控制者表示将一个指令放入数据线中
12		SHIELD	外壳屏蔽线
13	数据	D105	数据线 5
14	数据	D106	数据线 6
15	数据	D107	数据线 7
16	数据	D108	数据线 8
17	管理	REN	远程使能。设备示意其处于远程控制状态而非通过其控制面板进行控制
18		GND	地线/公共（与 DAV 组成双绞线）
19		GND	地线/公共（与 NRFD 组成双绞线）
20		GND	地线/公共（与 NDAC 组成双绞线）
21		GND	地线/公共（与 IFC 组成双绞线）
22		GND	地线/公共（与 SRG 组成双绞线）
23		GND	地线/公共（与 ATN 组成双绞线）
24		GND	信号地线

8 位并行数据总线一次能够发送一个 8 位字节的数据。每传输一个字节，总线就完成一个握手周期。总线中的每台设备都有自己的地址。将监视线（ATN）设置为低电平以表示来自控制者的指令。然后，将地址放入数据线中，把指令发给各台设备。设备地址以一个 7 位字的方式在数据线中传输，低 5 位是数据地址，其他两位是控制信息。如果这两位都是 0，那么指令将被发送到所有地址。如果第 6 位是 1，第 7 位是 0，那么被寻址的设备将被切换为“听者”。如果第 6 位是 0，第 7 位是 1，那么设备将被切换为“讲者”。

握手使用 DAV、NRFD 与 NDAC 这 3 条线，以保证“讲者”只有在“听者”正在侦听时讲话（见图 15-12b）。当“听者”准备好接收数据时，NRFD 线置为高电平。当数据放到总线中时，DAV 线置为低电平以通知设备数据已经准备好。当设备接收一个数据单词后，NDAC 线置为高电平以表示它已经收到该数据，NRFD 线置为低电平以表示它现在没准备好接收数据。当所有“听者”都将 NDAC 线置为高电平时，“讲者”将取消数据有效信号，并将 DAV 线置为高电平。这导致 NDAC 线置为低电平。为了将下一个单词放入总线，可以重复以上整个过程。

391~392

GPIB 总线以插入卡的方式（见图 15-13）用于为各种仪器提供连接到计算机的接口，例如数字万用表与数字示波器。插入卡与仪器之间通过标准电缆进行连接。

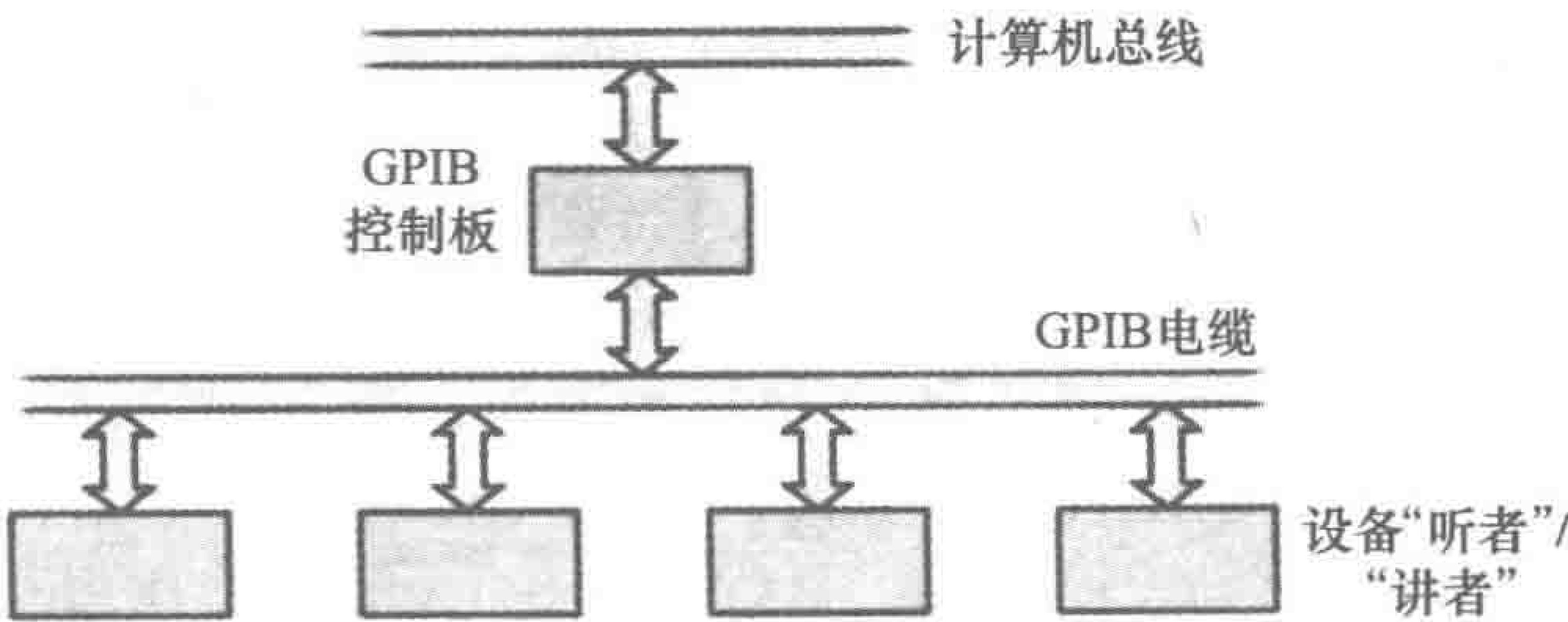


图 15-13 GPIB 硬件

15.7.1 其他总线

用于连接 CPU 与输入/输出端口或者其他设备的总线包括以下这些：

1）XT 计算机总线出现于 1983 年，用于 IBM PC/XT 及其兼容计算机，可以实现 8 位数据传输。

2）不久之后出现了 AT 总线，也称为工业标准体系结构（ISA）总线，用于使用 80286 和 80386 微处理器的 IBM PC 及其他兼容计算机的 16 位数据传输。AT 总线与 XT 总线兼容，所以在 AT 总线槽内可插入 XT 板。

3）扩展工业标准体系结构（EISA）总线用于使用 80386 和 80486 微处理器的 IBM PC 及其他兼容计算机的 32 位数据传输。

4）微通道体系结构（MCA）总线是 16 位或 32 位数据传输总线，用于 IBM 个人系统/1（PS/2）计算机。采用该总线的主板无法与 PC/XT/AT 主板兼容。

5）NuBus 网络用户总线是 32 位总线，用于 Apple 公司的 Macintosh Ⅱ型计算机。

6）S-bus 是 32 位总线，用于 Sun 公司的 SPARC 工作站。

7）TURBOchannel 是 32 位总线，用于 DECstation 5000 工作站。

8）VME 总线是由摩托罗拉公司设计的用于基于 68000 微处理器的 32 位系统。该总线目前广泛用于其他计算机系统的仪器总线。

以上这些总线都称为**底板总线**，因为在底板（见图 15-14）上可以安装很多连接器，并且各种具有特定功能的印刷电路板，例如内存，也能够插在底板上。底板为其上的每块板卡提供数据、地址与控制总线信号，因此可以通过购买现成板卡很轻松地实现系统扩展。数据采集卡和仪器与其他外设的接口卡必须能够接入这类计算机总线。数据采集卡和仪器卡通常有多种型号可选，以兼容不同类型的计算机。

393

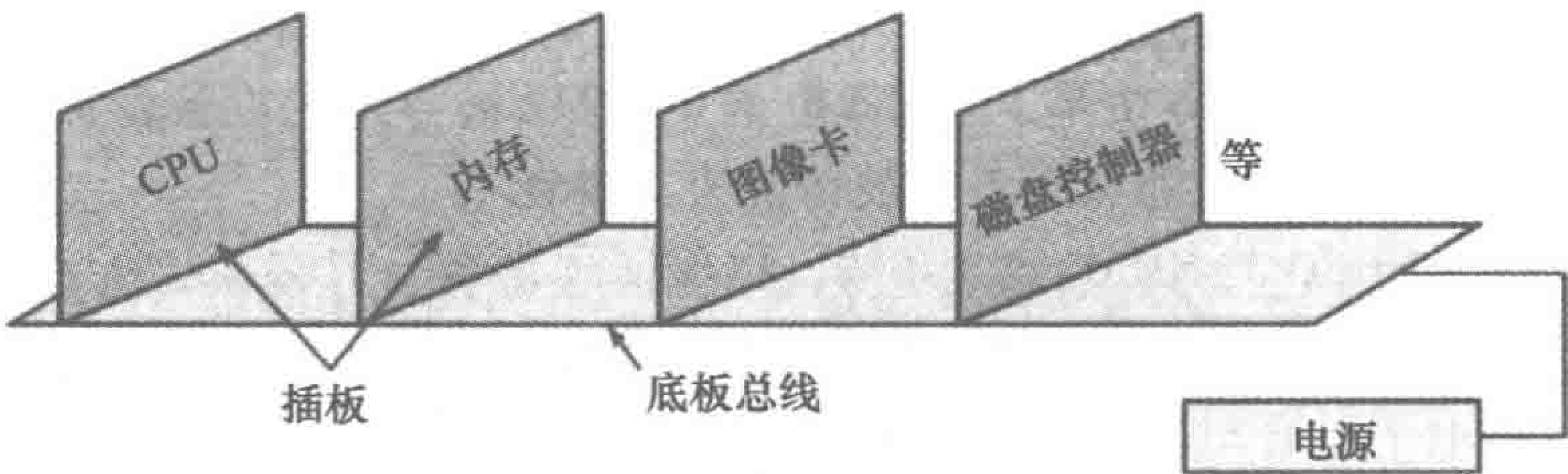


图 15-14　底板总线

VXIbus（VME 的仪器扩展）是 VME 总线规范的一个扩展，用于自动测试设备等仪器的应用场合，这类应用所需的更快的通信速度是 GPIB 总线无法提供的。VXI 总线还提供更好的同步与触发功能，并且是由许多仪器生产厂商组成的联盟所开发，因此不同公司的产品是相互兼容的。该系统需要将 VXI 主板连接到一台主机中。图 15-15 展示了多种可行的系统配置。在图 15-15a 中，一台 VXI 主机通过 GPIB 连接到一台外部控制器（一台计算机）。外部控制器使用 GPIB 协议与 VXI 机箱中的接口板进行通信，该接口板将 GPIB 协议翻译为 VXI 协议。这使得外部控制器能够将 VXI 仪器看作 GPIB 仪器，并用 GPIB 的方法对其进行编程。图 15-15b 展示了一台完整的计算机嵌入 VXI 机箱的方案。该方案提供了可能的最小系统外形尺寸，并允许计算机直接使用 VXI 背板总线。图 15-15c 使用了一条特殊高速系统总线电缆（MXI 总线）来连接计算机与 VXI 机箱，MXI 总线比 GPIB 总线快 20 倍。

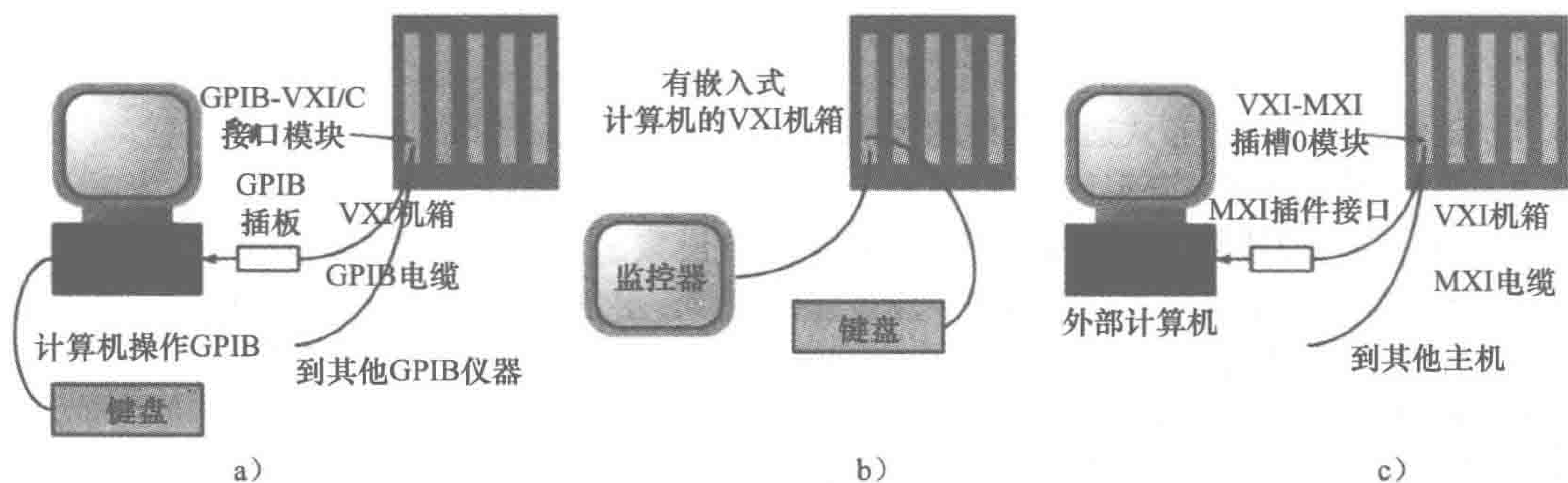


图 15-15 VXI 选择

15.8 无线协议

IEEE 802.11 是无线局域网的推荐标准，对网络的物理（PHY）层与介质访问控制（MAC）层进行了规范。MAC 层指定了一个载波侦听多路访问/冲突避免（CSMA/CA）协议。当一个节点有数据包准备发送时，它首先侦听线路以确保没有其他节点正在发送数据，如果线路空闲则发送；否则，它将等待并尝试再次发送。当发送数据包时，发送节点首先发送一个有数据包长度信息的准备发送（RTS）包，然后才发送数据包。当数据包被成功接收时，接收节点会发送一个确认（ACK）包。

394

蓝牙是一个短距离无线电传输的全球标准。当两台具有蓝牙功能的设备之间的距离在 10 米以内时，就可以建立一个无线连接。该标准广泛用于移动电话与个人计算机。

本章小结

外部总线是一组信号线，其作用是将微处理器、微控制器、计算机与 PLC 互连起来，同时并将它们与外围设备连接起来。

集中式计算机控制利用一台中央计算机控制整个工厂。在**分层式系统**中，针对所完成任务的不同，各个计算机处于不同的层次中。在**分布式系统**中，每个计算机系统都完成与其他计算机系统本质上相似的任务。

数据通信可以由**并行**或**串行传输**链路实现。串行数据传输可以是异步的或同步的。**异步传输**表明发送端与接收端的计算机不同步，各自有自己独立的时钟信号。**串行数据传输**有以下三种方式：单工方式、半双工方式与全双工方式。

网络是指一个允许两个或两个以上互联的计算机/微处理器进行数据交换的系统。常用的结构有如下几种：数据总线结构、星形结构、分层/树形结构、环形结构、网状结构。网络访问控制是必要的，用于保证任何时刻只有一个用户能够发送数据。对于基于环形结构的网络，采用令牌传递方式与时间片传递方式，对于总线或树形拓扑结构的网络，则采用带冲突检测的载波侦听多路访问。**协议**是一组正式的规则，用于管理数据格式、定时、排序、访问控制与错误控制。

国际标准化组织（ISO）定义了一个 7 层标准协议系统，也就是**开放系统互联**（OSI）模型。

串行通信接口包括 RS-232 及其更新版本，I^2C 与 CAN。**并行通信接口**包括通用仪器总线（GPIB）等。

习题

15.1 解释集中式与分布式通信系统的区别。

15.2 解释总线/树形与环形网络的结构。

15.3 如果一个局域网各节点间的距离大于100m，应该选择总线还是环形拓扑结构?

15.4 如果需要一个多信道的局域网，应该选择宽带传输还是基带传输?

15.5 什么是 MAP 与 TOP?

15.6 解释什么是通信协议。

15.7 简要解释两种用于局域网的多路访问控制方式。

15.8 一个 M68HC11 微控制器是一个“听者”，通过一条 GPIB 总线连接到一个“讲者”。指出完全握手所需要完成的连接。

395 ~ 396

15.9 为了使 M68HC11 微处理器的串行数据通信接口能够通过 RS-232 接口发送数据，需要解决什么问题?

15.10 什么是底板总线?

第16章　故障检测

本章目的

通过本章的学习，读者将能够：

- 了解微处理器系统中用于检测硬件和软件故障的技术方法。
- 了解仿真和模拟的使用。
- 了解 PLC 系统中故障检测的方法。

16.1　故障检测技术

本章简要介绍了在测量、控制及数据通信系统故障检测过程中需要解决的问题。具体系统或部件的故障检测技术细节，需要参照制造商的使用手册。

一般的故障检测方法包括：

1. 重复性检测

重复性检测包括复制或重复行为及比较重复过程得到的结果。在无故障的情况下，我们认为重复过程得到的结果应该是相同的。对于随时可能出现的错误，这可能意味着只需要重复操作两次，并比较两次结果；也可能意味着需要复制另一个系统，并比较由两个系统所产生的结果。这种方法的代价通常都很高。

2. 期望值检测

软件错误通常是通过输入具体的数值并检查是否得到期望值来检测的。如果没有得到预期值，那么软件就存在故障。

3. 时序检测

这主要是在指定时间内检测某个功能是否能够实现。这些检测通常涉及看门狗定时器。例如，使用可编程逻辑控制器（PLC）时，当一个操作启动时，一个定时器也会启动，如果在指定的时间内操作没有完成，那么就认为已经发生一个错误。看门狗定时器启动，发出警报并关掉部分或整个设备。

4. 反向检测

当输入和输出值之间有直接关系的时候，且输出值可以得到，那么引起输出值产生的输入值可以计算出来。然后，它可以与实际的输入相比较来判断是否存在故障。

5. 奇偶校验与误码检测

这种检测方式通常用于校验内存和数据传输错误。通信信道经常受到干扰，这可能会影响数据传输。为了检测数据是否已被损坏，将奇偶校验位添加到所传输的数据字中。使用奇偶校验位是为了使得这组数中“1”的总个数是奇数（奇校验）或者偶数（偶校验）。如果是奇校验，那么在传输后这个字会受到检测来确定“1”的个数是不是仍然是奇数。其他检测方式包括向所发送数据中添加代码来检测已损坏的数据位。

6. 诊断检测

诊断检测用来测试系统中组件的性能。给组件确定输入，将实际输出值与理论值进行比较，确定是否出现故障。

16.2　看门狗定时器

看门狗定时器基本上是一个超时前系统必须进行复位的定时器。如果定时器没有及时复位，那么就认为错误已经产生了。

图 16-1 是看门狗定时器的例子，这是一个简单的梯形图程序。它在 PLC 上用了一个看门狗定时器，用于监测活塞在气缸中的运动操作。当启动开关关闭时，电磁铁 A+ 得电，活塞在气缸中开始运动，同时启动定时器。当活塞运动到气缸底部的时候，限位开关 a+ 打开，定时器停止。但是，如果定时器溢出时，a+ 仍然没有打开，那么定时器的触点闭合，就会启动警报。因此，如果定时器设为 4s，则就是假定在此段时间内活塞可以到达气缸底部。但是，如果活塞卡死，在指定时间内没有到达气缸底部，那么警报器就会发出警报。

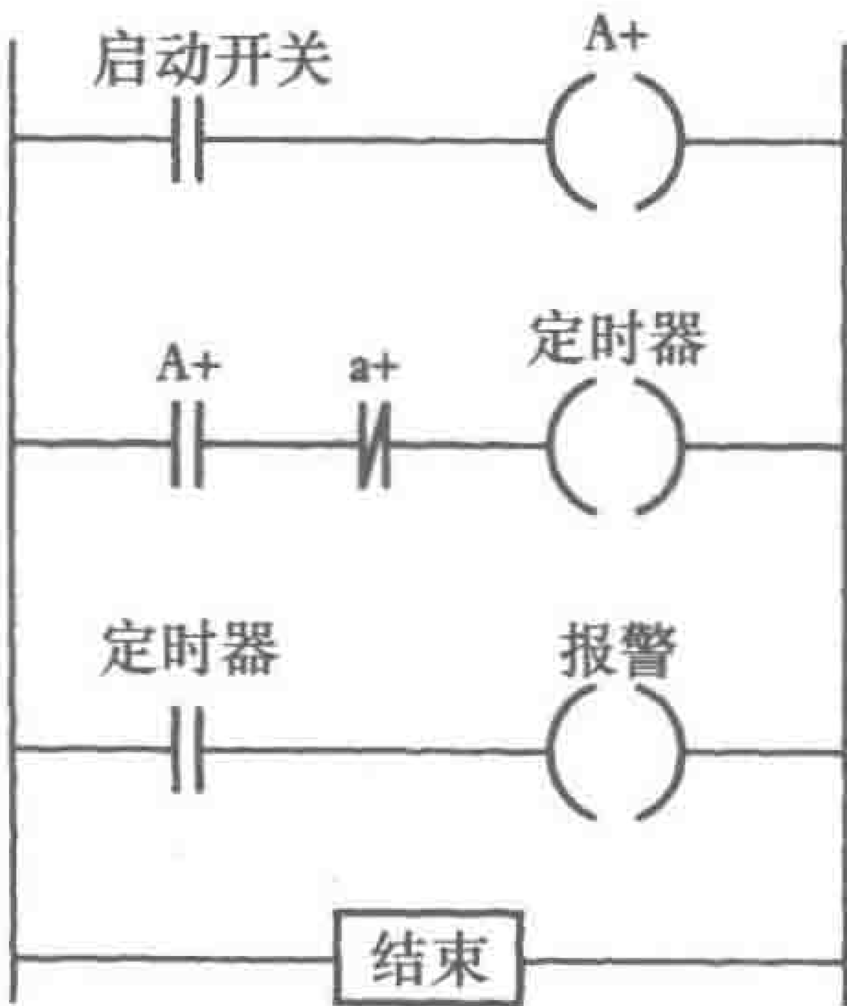


图 16-1　看门狗定时器

当微处理器执行内存中的指令时，附近的干扰可能随时会使数据总线上的数据反转。这样就会得到错误的字节。而软件缺陷也可能会导致子程序返回错误，引发处理器故障。这些故障会导致系统崩溃，可能给微处理器所控制的驱动器带来非常危险的后果。为了避免关键系统发生这种情况，看门狗定时器用来定时复位微处理器。

以具有看门狗定时器的微处理系统单片机 MC68HC11 为例。它内部具有看门狗定时器，称为**计算机操作正常**（Computer Operating Properly，COP）定时器，用来检测软件处理错误。当 COP 定时器启动时，主程序需要在 COP 溢出前定时复位它。如果在没有设置为重新开始计时前，看门狗定时器溢出，那么 COP 就不能正确复位。可以通过给地址为 $103A（0x103A）的 COP 复位寄存器（COPRST）写入 $55（在 C 语言中是 0x55）来对 COP 定时器进行复位操作，之后在程序中写入 $AA 来给 COP 定时器清零。如果程序始终在这两条指令之间执行，那么 COP 定时器就会溢出，导致执行 COP 错误复位程序。在汇编语言中，程序如下：

```
LDAA      #$55        ;复位定时器
STAA      $103A       ; 给 COPRST 写 $55
                      ; 其他程序行
LDAA      #$AA        ; 定时器清零
STAA      $103A       ; 给 COPRST 写 $AA
```

398

COP 工作时间的设置是通过给地址为 $1039（0x1039）的 OPTION 寄存器中的 CR1 和 CR2 写入 0 或者 1 来实现的。例如，将 CR1 和 CR2 设置为 0 时，溢出时间就为 16.384ms，将 CR1 设置为 1 并将 CR2 设置为 0，溢出时间就为 262.14ms。

16.3　奇偶校验与误码检测

为了检测噪声何时会使数据信号损坏并产生错误，常常需要使用奇偶校验等错误检测技术。

4.5.2 节简要介绍了如何用奇偶校验来检测错误。具体的方法是将一个多余位加入信息中，若是偶校验，那么 1 的总个数就为偶数；若是奇校验，1 的总个数就为奇数。例如，在偶校验系统中，会在字符 1010000 的最高有效位前添加一个 0，也就是 01010000，而在奇校验的系统中则添加 1，也就是 11010000。

这种方法只能检测信息中只有一位出错的情况，而当信息中两位出错时，会导致奇偶信息不发生变化。例如在用奇校验时上面的字符出现了一位错误，假如说是第三位，那么这个字符会被检测为 1101100，奇偶校验位出错，但是如果该字符第二位也出现错误，那么 1101110 的

奇偶校验位就是正确的。

如果没有检测到错误，系统就会向发送端返回字符 ACK，认为信号是正确的。如果检测到错误，就返回信号 NAK。这叫作**自动重复请求**（Automatic Repeat Request，ARQ）。NAK 信号会使信息重新传输。

可以通过**块奇偶校验**来提高错误检测效率。消息分成若干块，每块的末尾都会加入一个块校验字。例如在下面的块中，偶校验的校验位放在每行的末尾，而进一步的校验位放在每列的底部。

	信息位			校验位	
第一块字符	0	0	1	1	0
第二块字符	0	1	0	0	1
第三块字符	1	0	1	1	1
第四块字符	0	0	0	0	0
块校验位	1	1	0	0	0

在接收端对每行每列进行奇偶校验，通过包含有错误校验位的行列交叉点，可以检验每一位上的错误。

误码检测的另一种形式是**循环冗余校验**（CRC）。在传输终端，采用模 2 算法，用表示传输数据的二进制数除以一个预定义的数，所得到的余数就是随数据一起传输的 CRC 字符。在接收端，用数据和附加的 CRC 字符除以同一个数。如果没有发生传输错误，就没有余数。

399

一种常见的 CRC 码就是 CRC-16，它有 16 位校验码。它把这 16 位二进制数看成一个多项式的系数，同时位的个数等于多项式的最高次幂。首先用多项式的最高次幂，也就是说，x^{16}乘以这个数据块，然后再使用模 2 算法（在多项式中，也就是 $x=2$），除以 CRC 多项式

$$x^{16}+x^{12}+x^{5}+1$$

CRC 多项式是 10001000000100001。除以这个多项式的余数就是 CRC。

如下例，假设我们有数据 10110111 或者多项式

$$x^{7}+x^{5}+x^{4}+x^{2}+x^{1}+1$$

CRC 多项式为

$$x^{5}+x^{4}+x^{1}+1$$

或者 110011。数据多项式首先与 x^{5} 相乘，得到

$$x^{12}+x^{10}+x^{9}+x^{7}+x^{6}+x^{5}$$

也就是 1011011100000。再除以 CRC 多项式，得到

```
               11010111
110011 |1011011100000
        110011
         111101
         110011
          111010
          110011
           100100
           110011
             101110
             110011
              111010
              110011
               01001
```

余数为101010，那么它就是CRC码，并随数据一起传输。

16.4 常见的硬件故障

下面是在具体的组件和系统中比较常见的一些故障。

16.4.1 传感器

如果测试系统有故障，那么很可能是传感器出现了故障。可以做一个简单的测试，用一个新的传感器代替原来的，观察它对系统结果的影响。如果结果发生了变化，那么很可能是原来的传感器出现了故障；如果结果没有发生变化，那么故障就在系统的其他地方。也可以检测电压/电流源是否提供正确的电压/电流，连接线中电信号是否连续，传感器是否按照在制造商数据表中指定的情况下正确安装和使用等。

400

16.4.2 开关和继电器

开关触点之间废料污垢和颗粒是机械开关不能正常工作的一个常见原因。当触点断开的时候，跨过开关两端的电压表应该显示外加电压，当触点闭合的时候，显示的数据应该非常接近零。机械开关通常用来检测物品的位置，例如传送带上工件是否出现，但是不能够对工件是否对齐、启动杆是否弯曲做出正确响应。

检查继电器，如果有产生电弧或者接触焊弧的征兆，那么这个继电器就应该更换。如果继电器不能正常工作，那么就应该检查线圈两端的电压，如果电压正确，那么就用欧姆表检查线圈的通断。如果线圈两端没有电压，则故障很可能发生在继电器的晶体管开关上。

16.4.3 电动机

直流和交流电动机的维护都需要正确的润滑。直流电动机的电刷会磨损，需要更换，并且应按照制造商的说明更换。当单相电容启动交流电动机非常缓慢时，可能需要更换一个新的启动电容器。三相异步电动机没有电刷、换向器、滑环或启动电容器，所以除了严禁超载，唯一的要求就是定期维护和润滑。

16.4.4 液压和气动系统

引起液压和气动系统故障的最常见原因是污垢。小颗粒的灰尘可能会损坏密封件，堵住孔口，引起滑阀堵塞等。过滤器应定期检查和清理，组件只能在清洁的环境拆除，并定期检查和更换油液。在电路中，测试回路最常见的方法就是在很多测试点上测试电压。同样，在液压和气动系统中，需要测试一些点的压力。密封件损坏会导致液压和气压缸泄漏，这是比较正常的，除此之外，当驱动缸体时如果系统压力发生下降，则可以通过更换缸体的密封件来解决。叶片式电动机中的叶片会受到磨损，不能与电动机壳体很好地密封，导致电动机功率的损失。这时可对叶片进行更换。软管、管材和管件的泄漏也是很常见的故障。

401

16.5 微处理器系统

微处理器系统中的常见故障有以下几种：

1. 芯片故障

芯片通常是非常可靠的，但是偶尔也会出现故障。

2. 无源元件故障

微处理器系统通常包含无源元件，如电阻和电容。它们中的任何一个出现错误都会引起系统故障。

3. 断路

断路一般会导致信号线或电源线的中断。引起这些故障的典型原因是虚焊或者焊接接头失效，印刷电路板上线路的损坏，接头的错误连接以及电缆断线。

4. 短路

由于过多的焊锡将印刷电路板上相邻的线路连接，使得电路板上本不该相连的点和点之间发生短路。

5. 外部干扰

外部引入的脉冲会影响系统的运行，因为它们可能被捕获并被认为是有效的数字信号。典型的外部干扰来源于供电电源出现的尖峰跳变。这种尖峰跳变可能来自同一个电源驱动的其他设备，以及系统中开关的打开或关闭状态的切换。在系统的供电电源中加入滤波器可以有效消除这种尖峰。

6. 软件故障

尽管对软件进行了很多的测试，但仍然可能会包含缺陷，在某种特殊的输入和输出条件下会引起故障。

16.5.1 故障检测技术

微处理器系统的故障检测一般包括：

1. 肉眼检查

仔细观察一个有故障的系统也许就能发现故障的来源。举个例子，一个集成电路板在底座上松动了或者板上的焊锡过多都可以通过观察来发现。

2. 万用表

微处理器使用万用表检测的效果是非常有限的，但还是可以通过万用表检测电路是否存在短路、断路和电源故障。

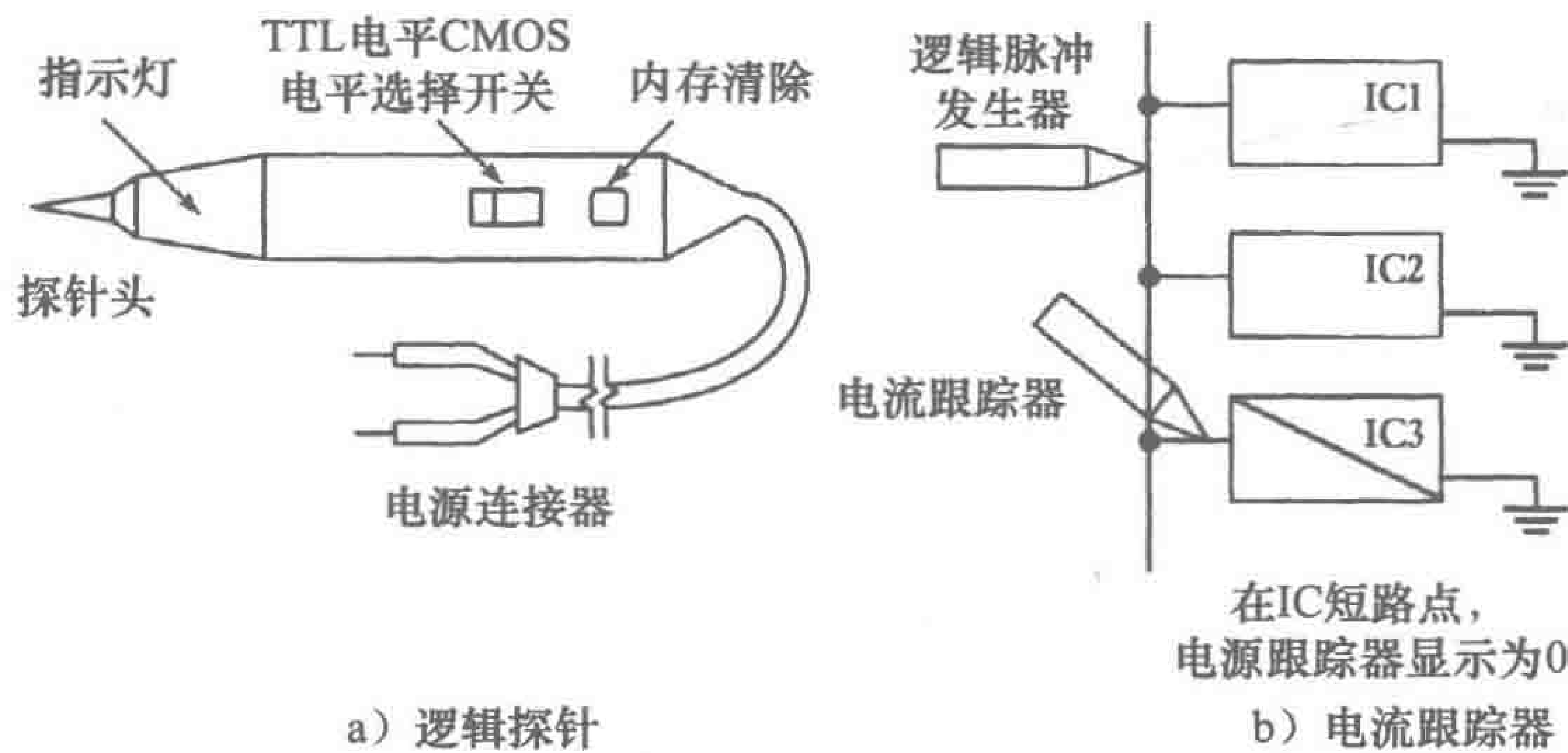
3. 示波器

示波器原则上更适用于测试周期信号，最明显的例子就是时钟信号。而微处理器发出的大多数信号是基于被执行程序的非周期信号。

4. 逻辑探针

逻辑探针是一种手持设备（见图16-2a），形似钢笔，可以用来检测电路上任何一点的逻辑电平。选择开关可以设置使用TTL电平还是CMOS电平。当探针尖端碰到有问题的触点时，指示灯显示它是否低于逻辑低电平0的临界值或者高于逻辑高电平1的临界值或者一个脉冲信号。探针中通常包括一个脉冲拓展电路用来拓展脉冲宽度以使指示灯的效果更明显。存储电路可以检测单一脉冲，按下存储清除按钮就会关闭指示灯并且所有关于逻辑检测的信息也会被一起清除。

402



a）逻辑探针　　b）电流跟踪器

图16-2　逻辑探针与电流跟踪器

5. 逻辑脉冲发生器

逻辑脉冲发生器是一种手持脉冲发生设备，形似钢笔。它可以输出可控的脉冲。将脉冲发

生器探针的尖端抵住焊点按下发生器上的按钮就能产生脉冲。经常用逻辑发生器的探针来检测逻辑门电路是否工作正常。

6. 电流跟踪器

电流跟踪器和逻辑脉冲发生器类似，但是它发射的是电流脉冲而不是电压脉冲。电流跟踪器对磁场非常敏感。因此可以用它来检测导体中由于脉冲电流产生的磁场变化情况。电流跟踪器的尖端沿着印刷电路板的线路移动就能找到电流流过的低阻抗的线路（见图 16-2b）。

7. 逻辑夹

逻辑夹是一个可以夹在集成电路芯片每个引脚上的测试设备。对应着每一个插针都有一个 LED 指示灯，这样每个插针的逻辑状态都能通过 LED 灯显示出来。

8. 逻辑比较器

逻辑比较器通过将集成电路与一个好的、参考的集成电路进行比较来测试集成电路（见图 16-3）。不需要将集成芯片从电路中取下，每个输入引脚都与参考的集成电路相对应输入引脚并联。同样，输出引脚也与参考集成电路的输出引脚并联。这两个输出用异或门电路相比较，如果这两个输出不相同，则此电路会输出相应的信号。脉冲展宽器用来展宽输入指示器信号的持续时间，这样就可以使短周期脉冲能够在指示器上以可观察到的周期显示出来。

403

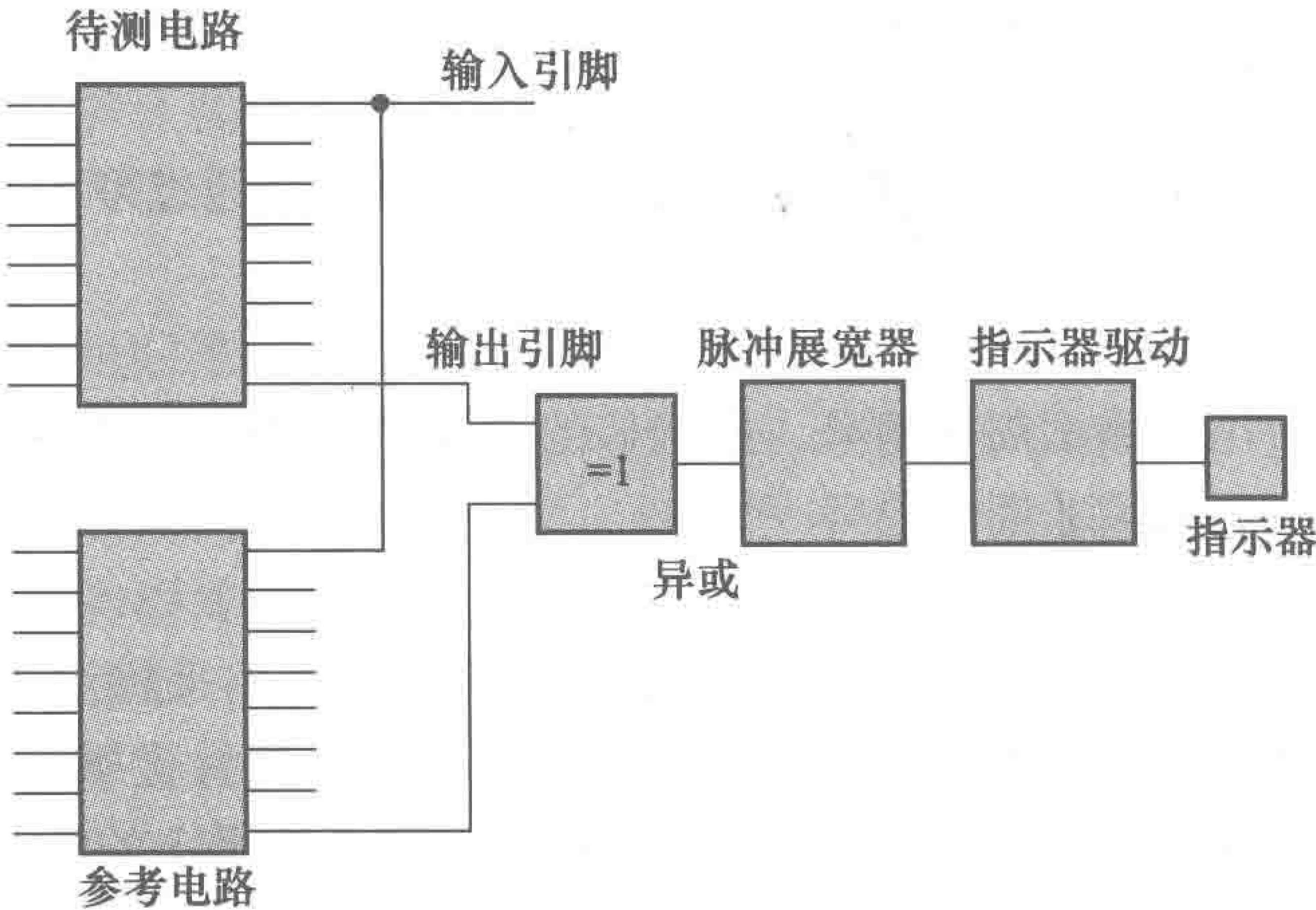


图 16-3 逻辑比较器

9. 特征分析仪

在模拟系统中，故障的发现通常包括通过电路示踪并检查各节点的波形，将这些波形和期望波形相比较来识别并确定故障的位置。在数字系统中，测量过程会更为复杂。因为这些节点的脉冲看起来都是一样的。为了分辨是否有故障，通常将脉冲转变为更易于识别的形式，一般称为**特征**，比如说 258F。把某一个节点获得的信号特征和其应该发生的情况做比较。当使用信号特征分析仪时，常常需要在设计该电路时使数据总线的反馈路径可以断开以易于测试，使得测试时错误的数字序列不被反馈进来。这样便可以使系统在测试时避免干扰反馈到总线。一个存储在 ROM 中的简短的程序能激活一个节点，并获取特征信号。如果数据总线被断开使其与内存隔离，那么微处理器自身就能进行检测，然后使其自由运行，并依次给每一个地址一个无操作指令。在这种状态下，微处理器总线的特征信号就能与那些预期值相比较了。

10. 逻辑分析仪

逻辑分析仪是对总线和装置的控制信号的逻辑电平进行采样，同时存储在先进先出（FIFO）寄存器中用于测试仪器。程序中在某一点上的数据捕获的开始和结束是通过使用“触

发字”来选择的。分析仪将它的触发字和引入的数据相比较，并且只有当触发字出现在程序中时才开始存储数据。数据捕获会一直持续下去，直到先前设定好的时钟脉冲停止。然后存储的数据以二进制、八进制、十进制或者十六进制的编码形式来存储，或者以时间为函数进行波形的显示，或者以助记符进行显示。

404

16.5.2 系统故障定位方法

下面是系统的故障定位方法：

1. 从输入到输出

将一个正确适当的信号输入到系统的第一个模块中，从第一个模块到最后一个依次连续地测量，然后重复执行，直到检测到故障模块。

2. 从输出到输入

将一个正确适当的信号输入到系统的第一个模块中，从最后一个模块到第一个依次连续地测量，然后重复执行，直到检测到故障模块。

3. 二分法

将合适的信号输入到第一个模块。将组成系统的模块分为两半。对每一半进行检测，确定是哪一半出现了故障。有故障的一半继续分成两半，然后重复进行这个步骤，直到找到故障模块。

16.5.3 自测试系统

在微处理器系统中，我们可以通过编写软件来建立正确运行的自测试程序。当系统第一次运行时，自测试程序会在主程序启动过程中完成初始化。例如，打印机的控制电路板里通常包括一个微处理器，而且其控制程序和测试程序一般都存储在 ROM 中。当第一次打开时，先运行这些测试程序，直到确认系统没有错误才开始接收数据并工作。

基本的 ROM 测试程序是将 ROM 中存储的所有字节进行计数并与已经存储的总和（就是所谓的**校验和测试**（checksum test））相比较。如果出现了不同，说明 ROM 出现了故障。如果数值相同，则认为没有故障。基本的 RAM 测试涉及存储数据的模式，即在 RAM 中相邻位取相反的逻辑电平，即将十六进制的55 和 AA 存入内存中的每一个位置，读出每一个存储的值并与发送的数据相比较检测其一致性（就是所谓的**棋盘格图样测试**（checker board test））。

16.6 仿真和模拟

仿真器（emulator）是用来测试微控制器及其程序的测试板。它包括：

1）微控制器。

2）用来存储数据和程序的微控制器的存储器芯片。

3）允许连接到被测试系统的输入/输出端口。

4）通信端口，其作用是使程序代码从计算机下载到微控制器并监控程序运行。

程序代码可以在主机上编写，并通过并行或串行方式下载到电路板的内存中。微控制器可以像程序在自身内存中一样来运行。表16-4 表示的是总体布置图。

405

微控制器的输入/输出线是通过板上的输入/输出端口与测试系统实现连接的。因此像微控制器直接插在它上面一样。这个电路板上已经编写了检测系统程序。这样就可以实现程序的运行监控，并可以检查和修改内存、输入/输出端口和相关寄存器的内容。

图16-5 给出了摩托罗拉 MC68HC11EVB 评估板的一些基本元素。它使用的监控程序称为“位用户快速友好逻辑运算帮助（Bit User Fast Friendly Aid to Logical Operations，Buffalo）”。一个8K 的 EPROM 存储着 Buffalo 监控程序。MC6850 异步通讯接口适配器（ACIA）（见13.5 节）与并行和串行线连接。RS-232 接口负责连接计算机和监控终端。

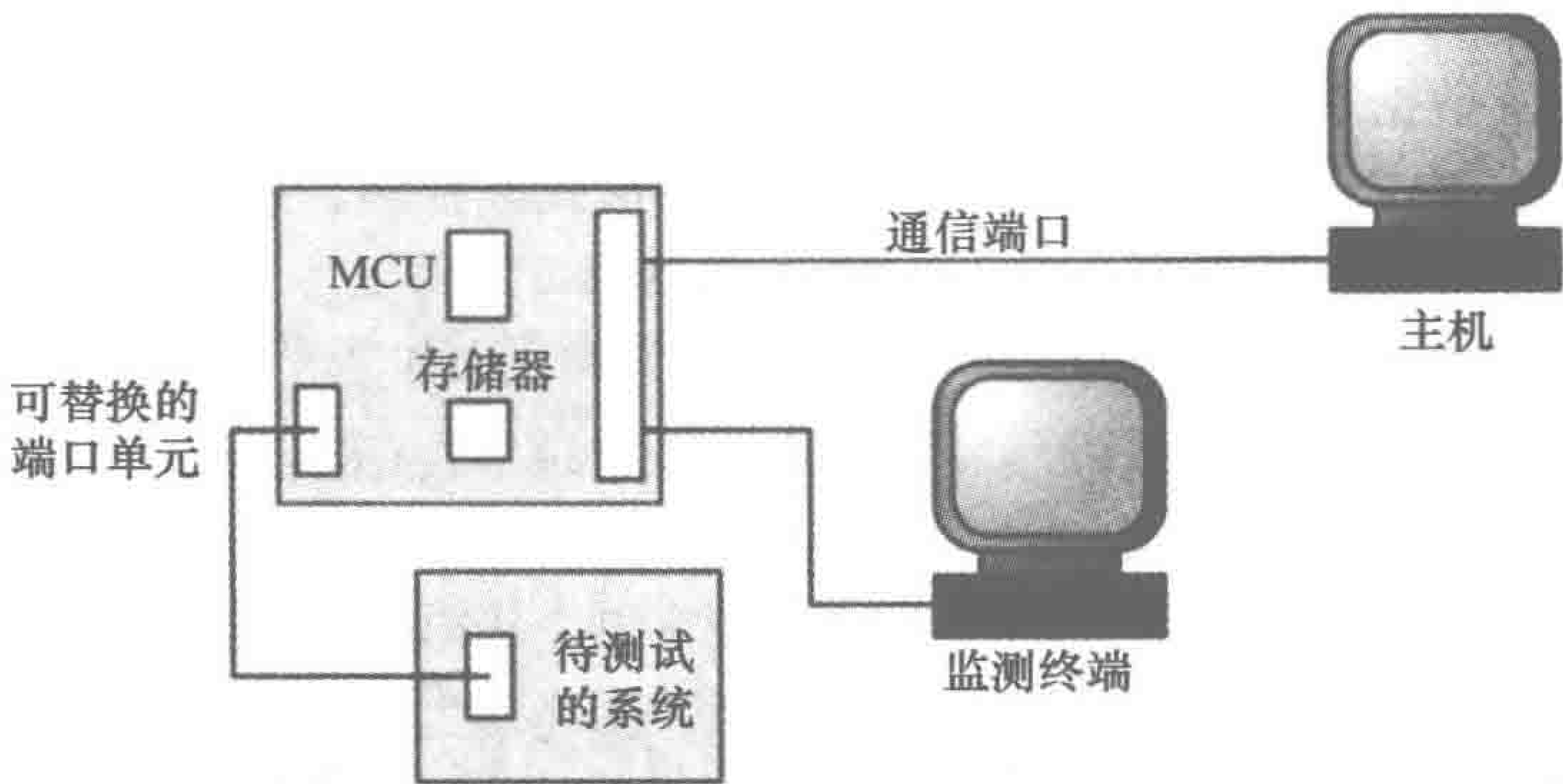


图 16-4　仿真器的使用

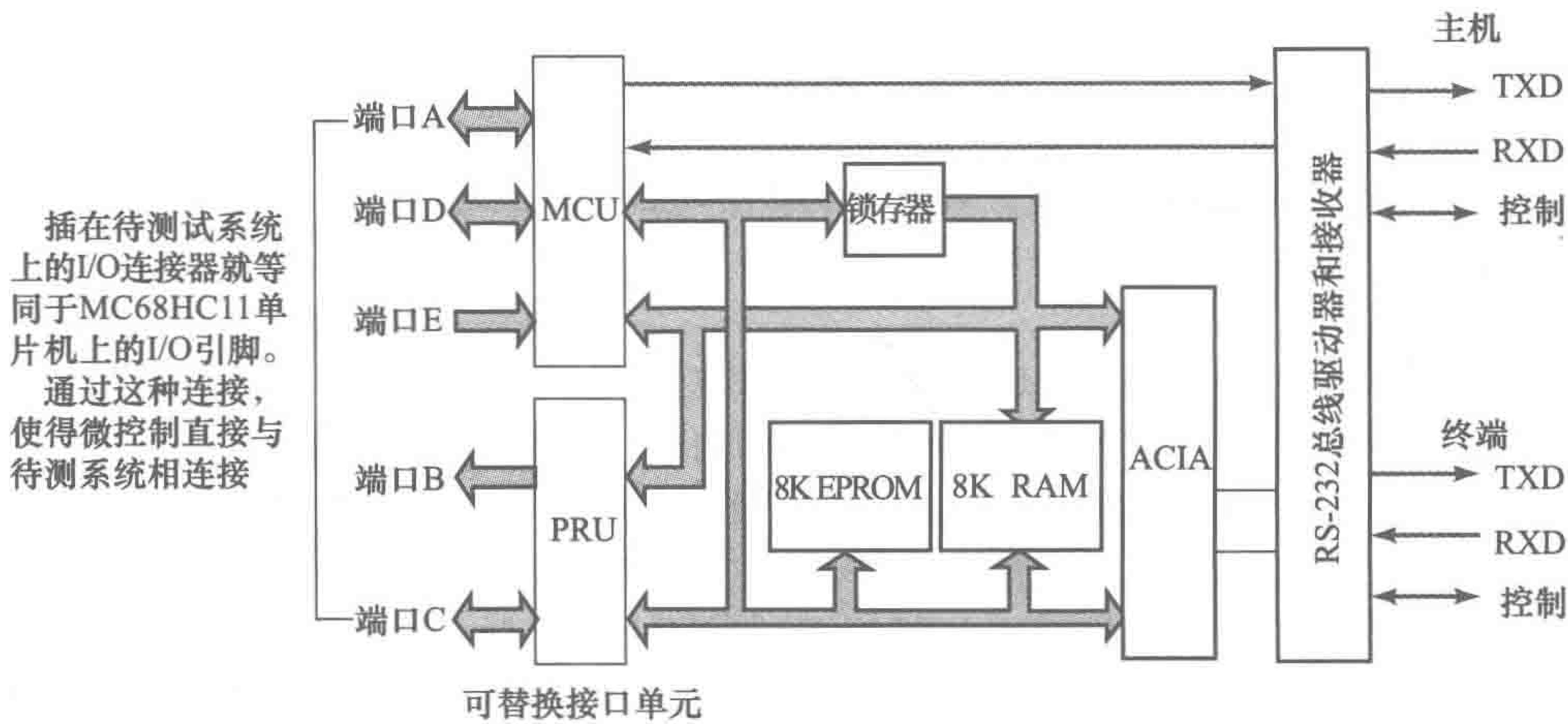


图 16-5　一款摩托罗拉单板机评估板

16.6.1　模拟

406

测试一个程序并不一定必须运行在实际的微控制器上，也可以通过运行计算机上的一个模拟微控制器程序来进行测试。这样的模拟能够帮助调试程序代码。显示屏被分为不同的窗口，用以显示不同的信息，比如说执行的源代码、指示当前状态的 CPU 寄存器和标志位、输入/输出端口、寄存器、定时器和内存的状态等。

16.7　PLC 系统

PLC 具有很高的可靠性。光隔离器或继电器使 PLC 自身与输入/输出端口上可能对其造成潜在破坏的电压或电流进行隔离。PLC 具有后备电池的 RAM 可以保护应用软件免受断电的损害。PLC 特殊的结构设计使它能够在工业现场环境中长时间可靠地工作。PLC 通常有一些内置的故障检测处理程序。严重故障会使 CPU 停止工作，其他不太重要的故障不影响 CPU 继续工作，但是会在显示屏上以故障代码的方式显示出来。当显示错误代码的时候，PLC 手册将会给出需要改正的提示。

16.7.1　程序测试

软件测试程序通过梯形图程序测试不正确的设备地址，并在屏幕上显示列表，或者打印出来。这个列表包括检测到的所有用到的输入点和输出点以及计数器和定时器设置等的错误信息。具体程序包括：

1）打开并显示相关的梯形图程序。

2）在屏幕上的下拉菜单中选择 Ladder Test。

3）然后屏幕会显示信息：Start from beginning of program（Y/N）?

4）输入Y，然后按 Enter（回车）键。

5）然后就会显示错误信息，或者显示“No errors found”。

例如，可能会有信息提示某个输出地址在程序中多次作为输出使用，用到的定时器或计数器没有预设值，计数器没有复位输入，没有 END 指令等。经过这样测试后，可能需要对程序进行更改。例如，按照屏幕上显示的一系列信息来对程序进行更改。

16.7.2 输入/输出测试

大多数 PLC 都有检测其输入和输出的所谓的强制设备。软件可以“强制”将输入和输出打开和关闭。PLC 强制控制输入和输出状态可以通过按下屏幕上的“强制”键或“监控”键来选择 FORCE 或者 MONITOR 模式来实现。强制进行输入时，我们可以检测相应输入的响应发生。存储好的程序开始运行，输入和输出开始模拟，因此可以检查所有预设的参数值。使用强制输入和输出需要特别谨慎，因为强制输出可能导致硬件以一种未知的危险方式运行。

407

图 16-6 给出了强制运行的显示类型示意图。我们可以看到输入从打开到关闭过程中，输出从不上电到上电是如何在一个梯形图中显示出来的。图 16-7a 是该梯形图的一部分。图 16-7b是当强制执行时的情况。图 16-7a 代表的是梯形图 11，X400、X401、M100 有输入，但是 X402 没有输入。这样，Y430 中就没有输出。对于梯形图 12，T450 定时器触点关闭，屏幕底部的显示指出没有多余的时间来运行 T450。因为 Y430 没有上电，Y430 的触点是断开的，所以 Y431 没有输出。如果我们现在强制从 X402 输入，那么屏幕上的显示就会发生变化，如图 16-7所示。Y430 上电，最后 Y431 也上电了。



图 16-6 监控模式标志

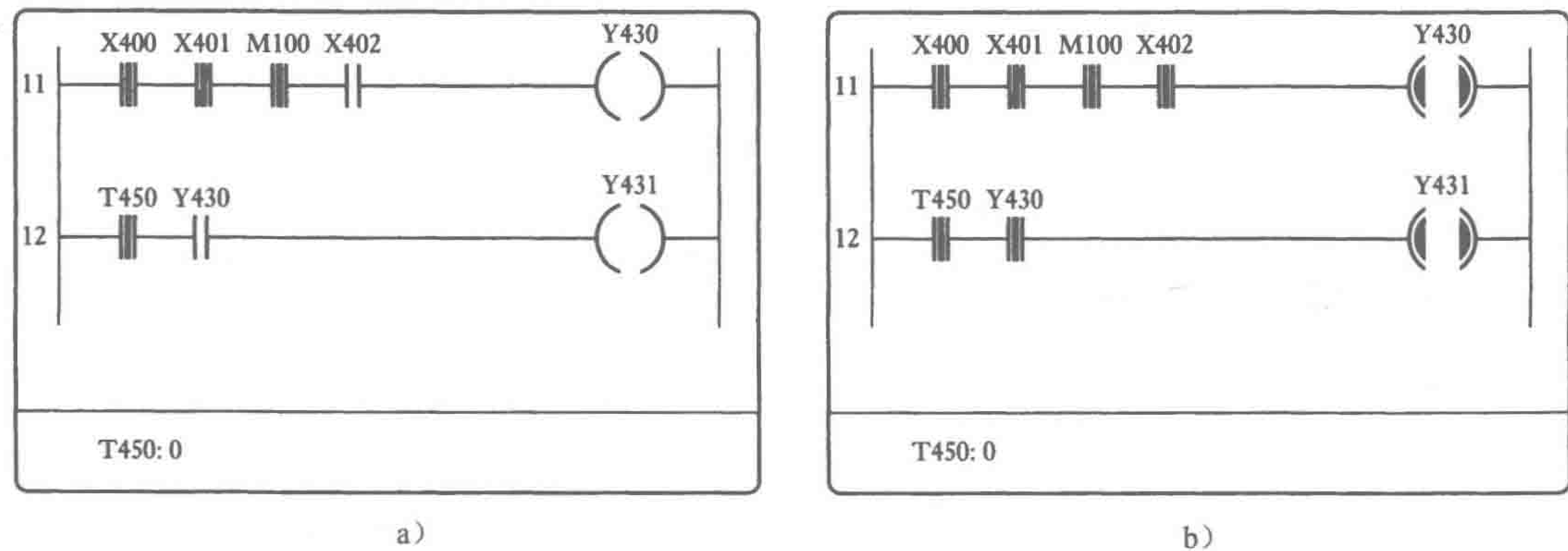


图 16-7 强制一个输入

16.7.3 PLC 作为系统的监视器

PLC 也可以用做控制系统的监视器。当输入值超出规定的限制，系统内部函数返回值大于、等于或小于预设值，或者操作时间比规定的时间长时，PLC 可以发出警报或者点亮红灯。图 16-1 就是用 PLC 梯形图实现看门狗定时器的某些操作的例子。

通常，在 PLC 系统中，通过状态灯来指示进程中所设定的最后输出。因此，如果有错误发

生，系统会停在那里。状态灯控制程序是内嵌在程序中的，所以每次有输出时相应的灯会被点亮，并关闭前一个输出所对应的状态灯。程序如图 16-8 所示。

408

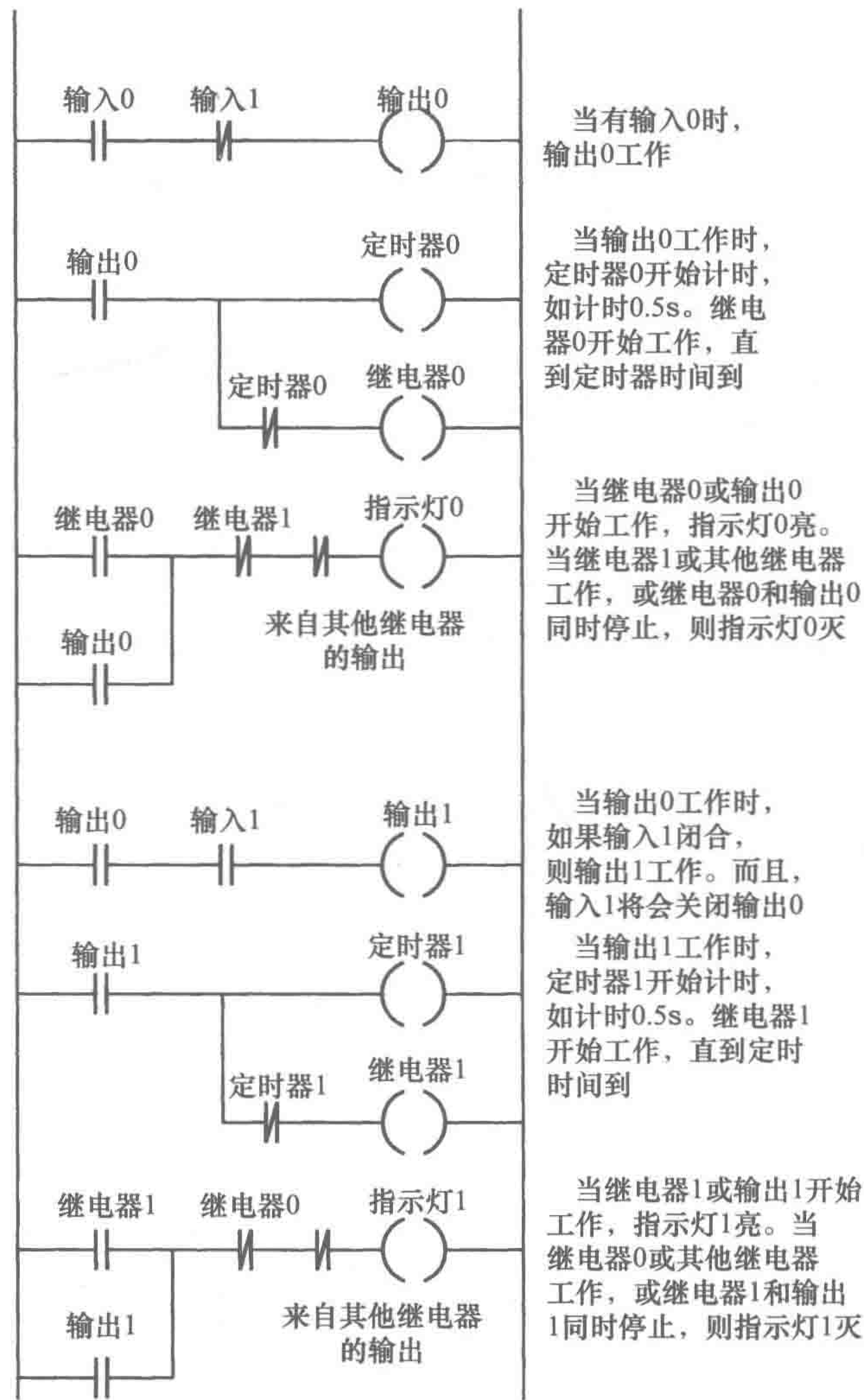


图 16-8　最终输出设置诊断程序

本章小结

故障检测技术包括重复检测、期望值检测、定时检测（例如看门狗定时器）、反向检测、奇偶和误码检测以及诊断检测。

看门狗定时器主要就是一个在溢出前系统必须进行复位的定时器。如果定时器不能及时复位，那么就认为错误已经发生。**奇偶校验**将一个多余位附加到信息中。在偶校验中，使 1 的总个数为偶数，奇校验中 1 的总个数为奇数。同时可以通过**块奇偶校验**来提高错误检测效率。消息分成若干块，每个块的末尾都会加入一个块校验字。**循环冗余校验**（CRC）采用模 2 算法，用表示传输数据的二进制数除以一个预定义的数，所得到的余数就是随数据一起传输的 CRC 字符。在接收端，用数据加上 CRC 字符除以同一个数，如果没有发生传输错误，就没有余数。

409

基于微处理器的系统可以使用自测试软件来检测工作是否正常。**仿真器**是用来检测微控制

器及其程序的测试板。检测一个程序也可以不用运行实际的微控制器，而是通过运行计算机上的一个模拟微控制器的程序来测试。

PLC 通常都有一些内置的故障检测程序。严重的故障会引起 CPU 停止工作，而不太严重的故障不影响 CPU 继续运行但屏幕上会显示故障代码。大多数的 PLC 都可以强制来检测其输入和输出状态。可以通过相关软件来实现“强制”控制输入和输出的通断状态。

习题

16.1 解释：(a) 重复性检测；(b) 期望值检测；(c) 反向检测；(d) 奇偶校验检测的含义。

16.2 解释在 PLC 控制的设备上如何使用看门狗定时器来指示故障存在。

16.3 解释单片机 MC68HC11 中 COP 的功能。

16.4 F2 系列 Mitsubishi PLC 具有如下诊断功能：
可编程检测（求和检测、语法检测、电路检测）、看门狗定时器、电池电压、电源电压。解释上述名词的定义。

16.5 解释微处理器系统中是如何使用自测试软件来检测它的 ROM 和 RAM 的。 410

第五部分　系统模型

第17章　基本系统模型

本章目的

通过本章的学习，读者将能够：

- 认识系统模型在预测系统行为时的重要性。
- 从基本模型推导出机械、电气、流体和热力系统的模型。
- 比较机械、电气、流体和热力系统模型的相似之处。

17.1　数学模型

首先来看一个例子。微控制器控制电动机的转动。电动机轴的转动和时间的关系是什么？转速不会马上达到设定速度，但在一段时间之后可以达到。再看另外一个例子，液压系统如何控制阀门使水箱里的水位随时间进行变化？水位不能马上达到设定值，只能在一段时间之后达到。

为了理解系统的行为，我们需要通过**数学模型**来进行描述。这些模型是一个真实系统在某些方面的简化表达。它们通过方程式来表示系统输入和输出之间的关系，然后可用于对指定条件下系统的行为进行预测，例如对于给定的一系列输入的输出，或者在特殊参数改变时的输出。为了得到系统的数学模型，需要先做一些假设和简化，并且在模型简化和真实行为的表达需求之间做出平衡。例如，我们可能对弹簧做出假设，伸长量 x 和输出力 F 之间成正比关系，从而得到数学模型 $F=kx$。在伸长量与力不成正比的时候，这个简单模型可能并不能精确地预测一个真实弹簧的行为，因为在大输入力的情况下弹簧可能受到永久性的损坏，甚至断裂，这就不能通过简单模型进行预测。

系统运行的基本物理原理是任何数学模型的基础。本章将介绍多种类型的系统，包括机械、电气、热力学以及流体系统的例子。

411
≀
413

像小孩子搭建房屋、汽车、吊车等玩具一样，一个系统能通过一些类似基本的模块组成。每个模块具有一个单独的性质或功能。以一个电路系统为例，一个电路可由代表电阻、电容和电感行为的模块组成。假设电阻模块只具有电阻的特性，电容模块只有电容特性，而电感模块只有电感特性。通过对这些模块进行不同的组合，能组成不同的电路系统，系统的输入/输出关系也能通过模块之间的适当组合来得到。通过这种方式组合成的系统称为**集总参数系统**。这是因为每个参数，如性质或功能，是单独考虑的。

在机械、电气、热力、流体系统中用到的模块的行为特征类似。本章主要介绍模块以及通过对模块的组合来获得实际物理系统的数学模型。第18章会研究更加复杂的模型。需要强调的是，这些模型只是辅助进行系统设计，真实的系统通常存在非线性特性，或者与这几章提出的理想模型有偏离。这个问题将在第18章深入讨论。

17.2 机械系统模块

用来表示机械系统模型的模块有弹簧、阻尼器和质量。**弹簧**代表系统刚度；**阻尼器**产生阻止运动的反向力，如摩擦或者阻尼效应；**质量**表示惯性或对加速度的阻抗（见图17-1）。虽然机械系统不一定是由弹簧、阻尼器和质量模块组成，但一定会具有刚度、阻尼和惯性的相关特征。所有这些模块都可以视为以力作为输入、位移作为输出的系统。

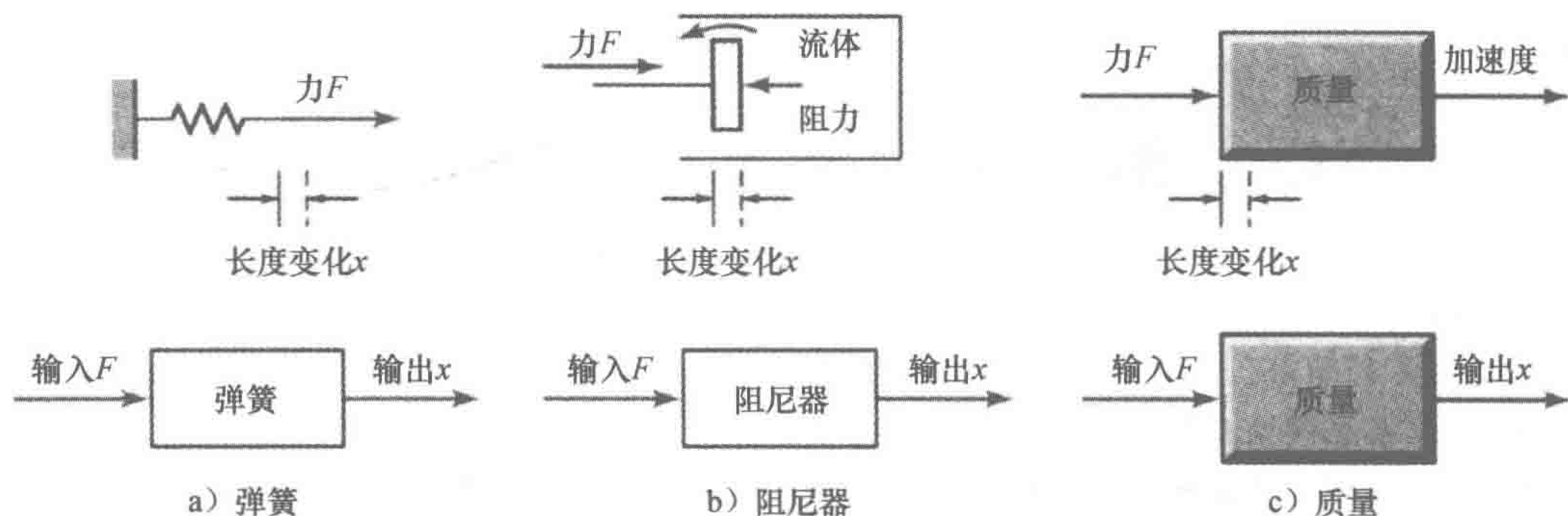


图17-1　机械系统

弹簧的刚度定义为拉伸或压缩弹簧的力 F 与拉伸或压缩长度 x 之间的关系（见图17-1a）。在输入力和伸缩量成正比的情况下，即一个线性弹簧，满足：

$$F = kx$$

414

其中 k 是一个常量。k 的值越大，则拉伸或者压缩弹簧所需的力越大，即弹簧的刚度越大。对弹簧施加力的物体也同样受到一个力，根据牛顿第三定律，这个力是弹簧的反作用力，方向相反，大小相同，即 kx。

阻尼模块表示在液体阻力中推动物体，或者克服摩擦力推动物体时所受到的力的类型。物体被推动的速度越快，产生的反作用力越大。阻尼模块用于形象地表示使物体减速的阻尼力，阻尼器可以表示为一个活塞在一个封闭的缸体中运动（见图17-1b）。活塞的运动使液体从气缸内的一侧流向另一侧，从而产生阻力。理想情况下，阻尼或阻力 F 正比于活塞速度 v，从而有

$$F = cv$$

其中 c 是常量，在一定速度下，c 的值越大，阻力越大。由于速度 v 是位移 x 的变化率，即 $v = \mathrm{d}x/\mathrm{d}t$，所以有

$$F = c\frac{\mathrm{d}x}{\mathrm{d}t}$$

从而活塞位移 x（输出）与阻尼力（输入）的关系和速度有关。

质量模块（见图17-1c）的特性为：质量越大，需要使其产生一定加速度的力越大。根据牛顿第二定律，力 F 和加速度 a 的关系为 $F = ma$，其中比例常量称为质量 m。加速度是速度的变化率，即 $\mathrm{d}v/\mathrm{d}t$，速度 v 是位移 x 的变化率，即 $v = \mathrm{d}x/\mathrm{d}t$。因此

$$F = ma = m\frac{\mathrm{d}v}{\mathrm{d}t} = m\frac{\mathrm{d}(\mathrm{d}x/\mathrm{d}t)}{\mathrm{d}t} = m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}$$

拉伸弹簧，使质量块产生加速度，以及移动阻尼器中的活塞都需要能量。对于弹簧和质量块能量可以恢复回来，但是阻尼器不能。弹簧在拉伸的时候存储能量，在弹回到原始位置的时候将能量释放出来。在伸缩距离 x 时，存储的能量为 $1/2kx^2$。由于 $F = kx$，从而有

$$E = \frac{1}{2}\frac{F^2}{k}$$

在质量模块中，以速度 v 运行的质量也存储了能量，这种能量称为动能，在停止运动时释放出来：

$$E = \frac{1}{2}mv^2$$

然而，阻尼器不能存储能量。它在没有输入力的情况下，不能返回到初始位置。阻尼器会消耗能量，其消耗功率 P 与速度有关，即 [415]

$$P = cv^2$$

17.2.1 转动系统

弹簧、阻尼器和质量块是机械系统的基本模块，它们包括力和直线运动，但不包含任何旋转运动。如果存在转动，则等价的三个基本模块为**扭转弹簧**、**旋转阻尼器**以及**转动惯量**，即旋转质量块的惯量。在这些基本模块下输入为转矩，输出为旋转角度。对于扭转弹簧，转动角度 θ 正比于转矩 T。因此，

$$T = k\theta$$

旋转阻尼器相当于一个圆盘在液体中旋转，受到的阻尼转矩 T 正比于角速度 ω，并且角速度是角度改变的速率，即 $\mathrm{d}\theta/\mathrm{d}t$，

$$T = c\omega = c\frac{\mathrm{d}\theta}{\mathrm{d}t}$$

对于转动惯量模块，为了产生一定的转动加速度 α，越大的转动惯量 I 则需要越大的转矩：

$$T = I\alpha$$

转动加速度是角速度的变化率，即 $\mathrm{d}\omega/\mathrm{d}t$，角速度是角位移的变化率，于是

$$T = I\frac{\mathrm{d}\omega}{\mathrm{d}t} = I\frac{\mathrm{d}(\mathrm{d}\theta/\mathrm{d}t)}{\mathrm{d}t} = I\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2}$$

扭转弹簧和转动惯量能存储能量，而旋转阻尼器只消耗能量。扭转弹簧在旋转 θ 角度时存储的能量为 $\frac{1}{2}k\theta^2$，其中 $T=k\theta$，有

$$E = \frac{1}{2}\frac{T^2}{k}$$

转动惯量模块以角速度 ω 旋转时的动能 E 为

$$E = \frac{1}{2}I\omega^2$$

旋转阻尼器以角速度 ω 旋转时的消耗功率 P 为

$$P = c\omega^2$$

表 17-1 总结了定义机械基本模块特性的方程。其中，对于直线运动（平动），输入为力 F，输出为位移 x；对于旋转运动，输入为转矩 T，输出为角位移 θ。 [416]

表 17-1 机械系统模块

模块	描述方程	能量存储或功率消耗
平移的		
弹簧	$F = kx$	$E = \frac{1}{2}\frac{F^2}{k}$

（续）

模块	描述方程	能量存储或功率消耗
阻尼器	$F = c\frac{dx}{dt} = cv$	$P = cv^2$
质量	$F = m\frac{d^2x}{dt^2} = m\frac{dv}{dt}$	$E = \frac{1}{2}mv^2$
旋转的		
弹簧	$T = k\theta$	$E = \frac{1}{2}\frac{T^2}{k}$
旋转阻尼器	$T = c\frac{d\theta}{dt} = c\omega$	$p = c\omega^2$
转动惯量	$T = I\frac{d^2\theta}{dt^2} = I\frac{d\omega}{dt}$	$E = \frac{1}{2}I\omega^2$

17.2.2 构建一个机械系统

很多系统可以认为本质上是一个质量块、一个弹簧和一个阻尼器的组合，如图 17-2a 所示。其中输入为力 F，输出为位移 x（见图 17-2b）。为了便于计算系统中力和位移的关系，可以认为系统只有一个质量块，并且输入力施加在质量块上。只有质量和作用在它上面力的图称为**受力图**（见图 17-2c）。

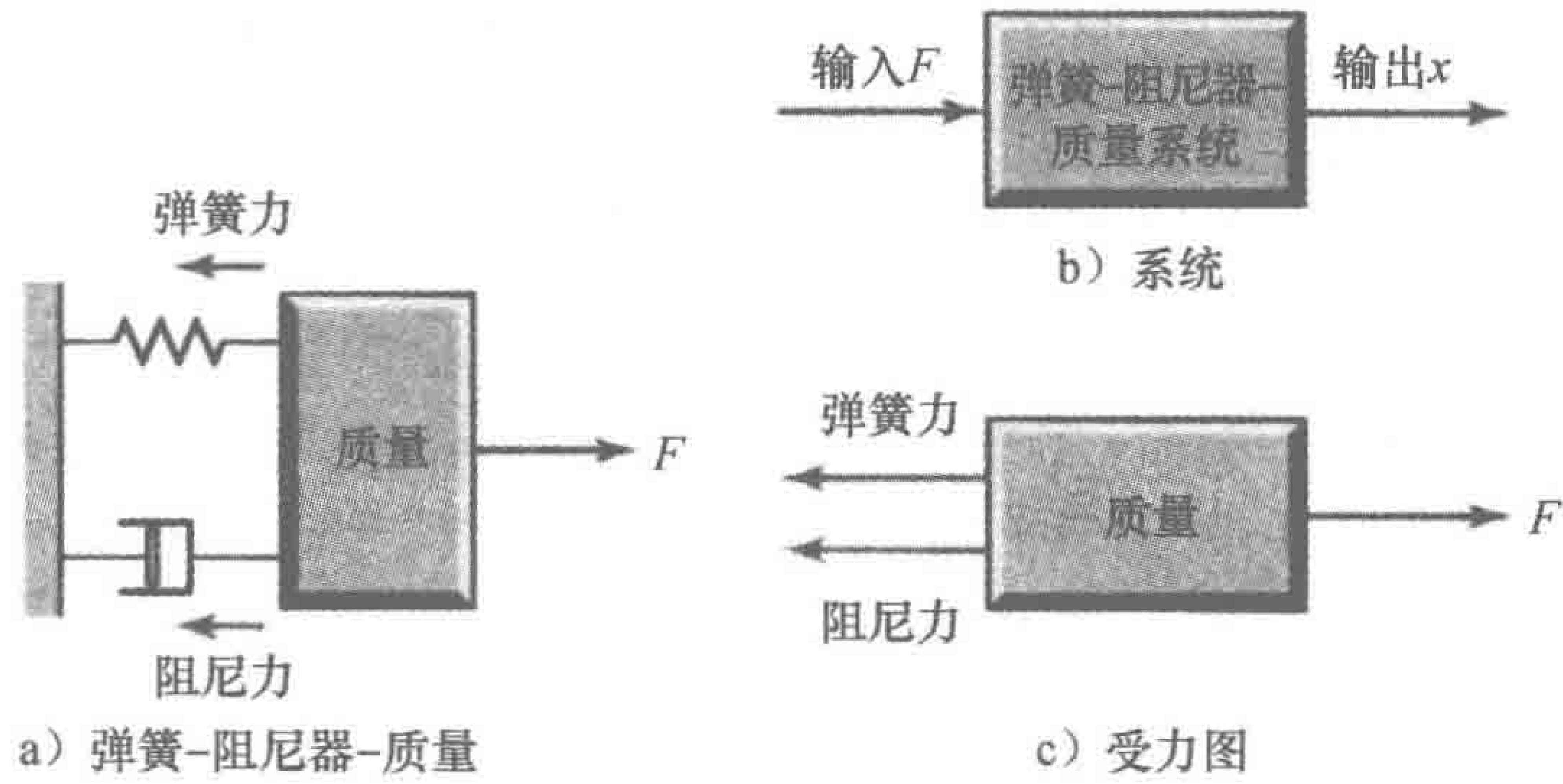


图 17-2 一个机械系统

当多个力同时作用在一个物体上时，它们的等价合力为每个力的向量和。如果力全部作用在同一条直线或者平行线，则作用在物体上的合力为代数和。对于图 17-2c 中的质量块，如果只考虑作用在模块上的力，则合力为施加力减去弹簧的张紧力或压缩力和阻尼器的阻力。因此：

417

$$\text{施加到质量块 } m \text{ 的净力} = F - kx - cv$$

其中，v 是阻尼器中活塞的速度。净力是施加到质量块上并引起质量块加速的力，因此有

$$\text{施加到质量块的净力} = ma$$

即

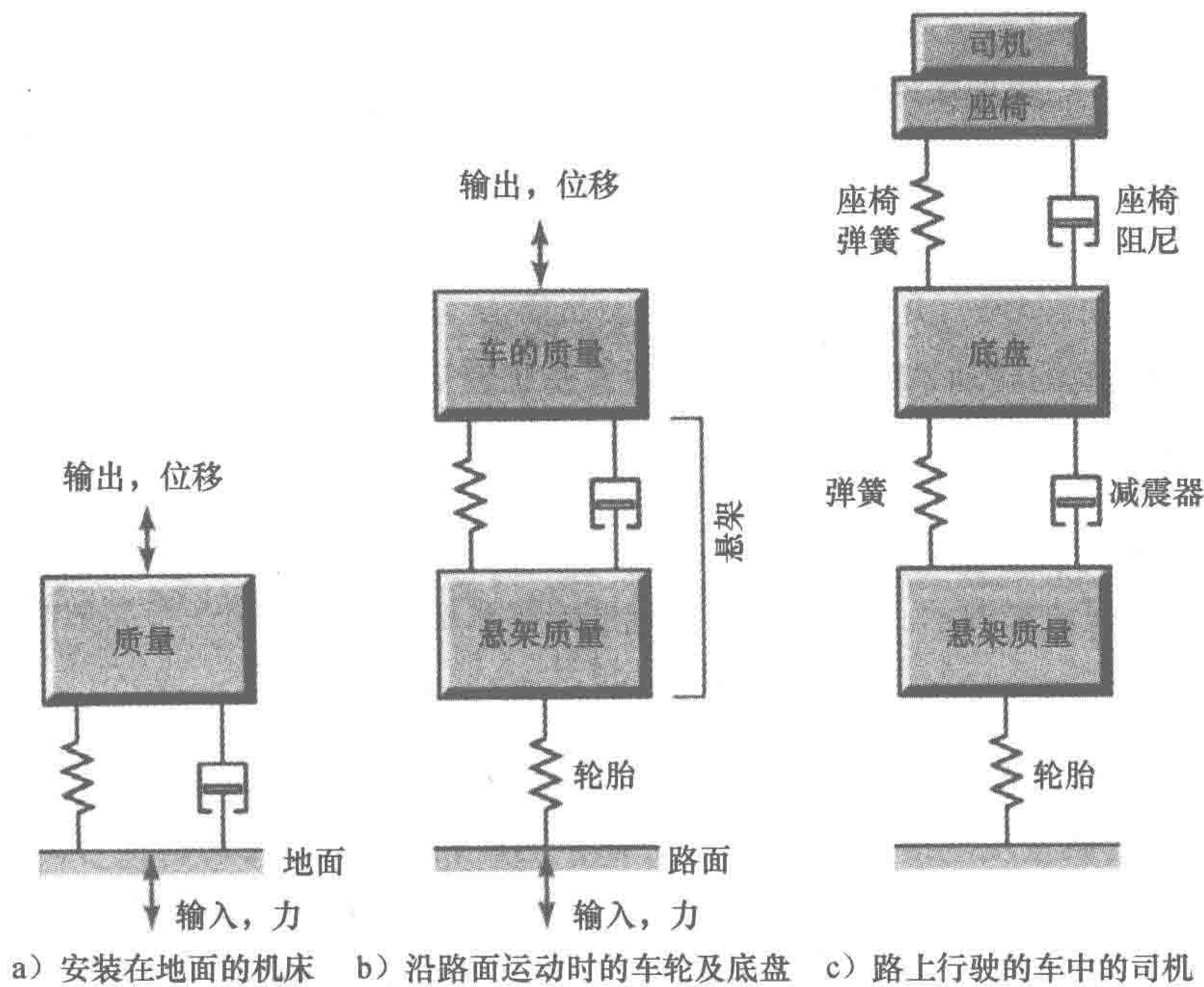
$$F - kx - c\frac{dx}{dt} = m\frac{d^2x}{dt^2}$$

重新排列后为

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+c\frac{\mathrm{d}x}{\mathrm{d}t}+kx=F$$

这个方程称为**微分方程**，它描述系统输入力 F 和输出位移 x 之间的关系。由于有 $\mathrm{d}^2x/\mathrm{d}t^2$ 项，所以方程称为二阶微分方程。一阶微分方程只包含 $\mathrm{d}x/\mathrm{d}t$。

很多系统能通过弹簧、阻尼器和质量块的组合来建立模型。图 17-3 说明了部分这样的系统。图 17-3a是一个安装在地面的机床模型，用来研究地面扰动对机床位移的影响。图 17-3b 为一个汽车或卡车的车轮及悬架系统模型，用来研究车子在不平整路面上运动时的表现，作为车辆悬架设计的基础。图 17-3c 展示了以上模型为组成部分的一个更大的模型，用来预测司机的舒适度。对这些模型的分析过程类似于简单的弹簧 - 阻尼器 - 质量块模型。每个质量块可以单独地画出其受力图，每个质量块上所受的合力就是该质量块的质量与其加速度的乘积。



a）安装在地面的机床　b）沿路面运动时的车轮及底盘　c）路上行驶的车中的司机

图 17-3　模型

下面以图 17-4 为例，推导描述输入力 F 和输出位移 x 之间的微分方程，说明以上原理。

质量块上的净力为 F 减去弹簧阻力，弹簧阻力为 k_1x 和 k_2x，有

$$\text{净力}=F-k_1x-k_2x$$

由于净力使质量块产生加速度，则

$$\text{净力}=m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}$$

因此

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+(k_1+k_2)x=F$$

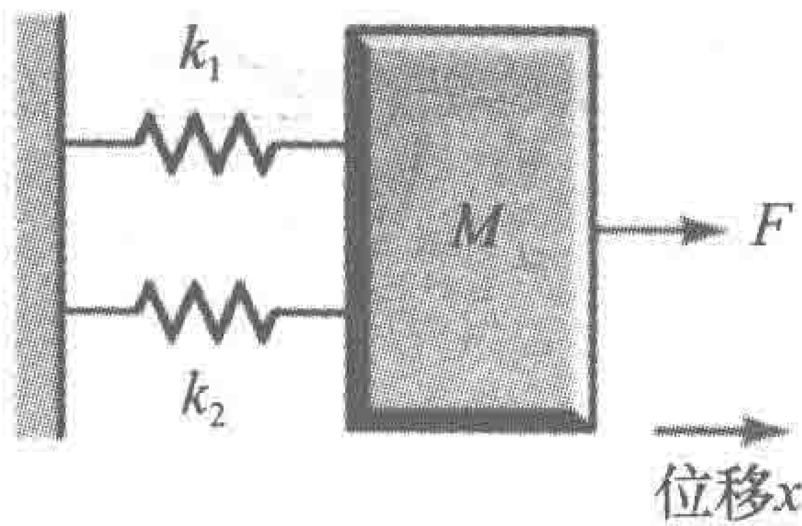


图 17-4　例子

对由多个部分组成的机械系统，获得描述输入/输出关系的微分方程可以按如下步骤进行：

1）将系统中的不同部分分离开，并画出每个部分的受力图。

2）根据每个部分的受力情况，写出它们的受力方程。

3）联立系统不同部分的方程从而得到系统的微分方程。

下面以图 17-5a 为例，说明在力 F 作用下质量块 m_1 运动微分方程的推导。画出受力图(见图 17-5b)。外力 F 和上面弹簧的伸缩力作用在 m_2 上，上面弹簧所产生的伸缩力与其伸缩长度 (x_3-x_2) 成正比，其大小为 $k_2(x_3-x_2)$。从而净力为

$$净力 = F - k_2(x_3 - x_2)$$

这个净力使得质量块加速，即

$$F - k_2(x_3 - x_2) = m_2 \frac{d^2 x_3}{dt}$$

418~419

根据 m_1 的受力分析图，其受到上面弹簧的力 $k_2(x_3-x_2)$ 和下面弹簧的力 $k_1(x_1-x_2)$，从而质量块的净力为

$$净力 = k_1(x_2 - x_1) - k_2(x_3 - x_2)$$

这个力导致 m_1 产生加速度，有

$$k_1(x_2 - x_1) - k_2(x_3 - x_2) = m_1 \frac{d^2 x_2}{dt}$$

因此我们得到两个联立的二阶微分方程来描述系统的行为。

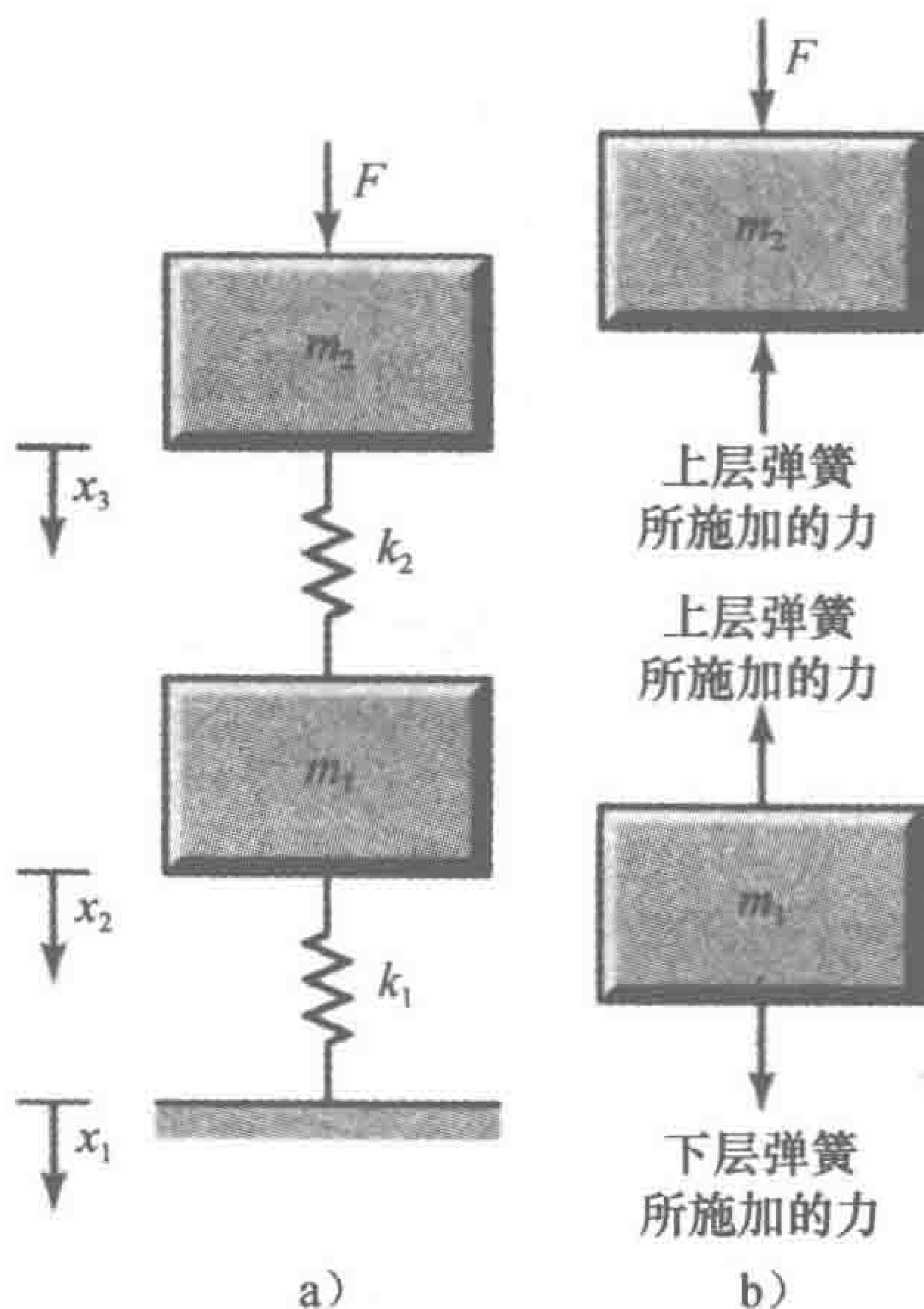


图 17-5　质量－弹簧系统

也可以构建类似的转动系统模型。为了计算系统转矩和角位移的关系，只考虑一个转动质量块，并且转矩施加在质量上。当有多个转矩时，它们的等价转矩为其代数和，并考虑它们的方向。图 17-6a 中的系统包含一个用于转动转轴末端质量的转矩，可以用图 17-6b中的转动模块来表示。与图 17-2 中线性位移系统比较，得到一个类似的方程：

$$I \frac{d^2\theta}{dt^2} + c\frac{d\theta}{dt} + k\theta = T$$

很多控制系统是电动机通过齿轮系带动负载运动。图 17-7 为这类系统的一个简单模型，由转动惯量为 I_1 的质量块连接齿数为 n_1 半径为 r_1 的齿轮 1，以及转动惯量为 I_2 的质量块连接齿数为 n_2 半径为 r_2 的齿轮 2 组成。一般情况下，忽略齿轮的转动惯量和转动阻尼。

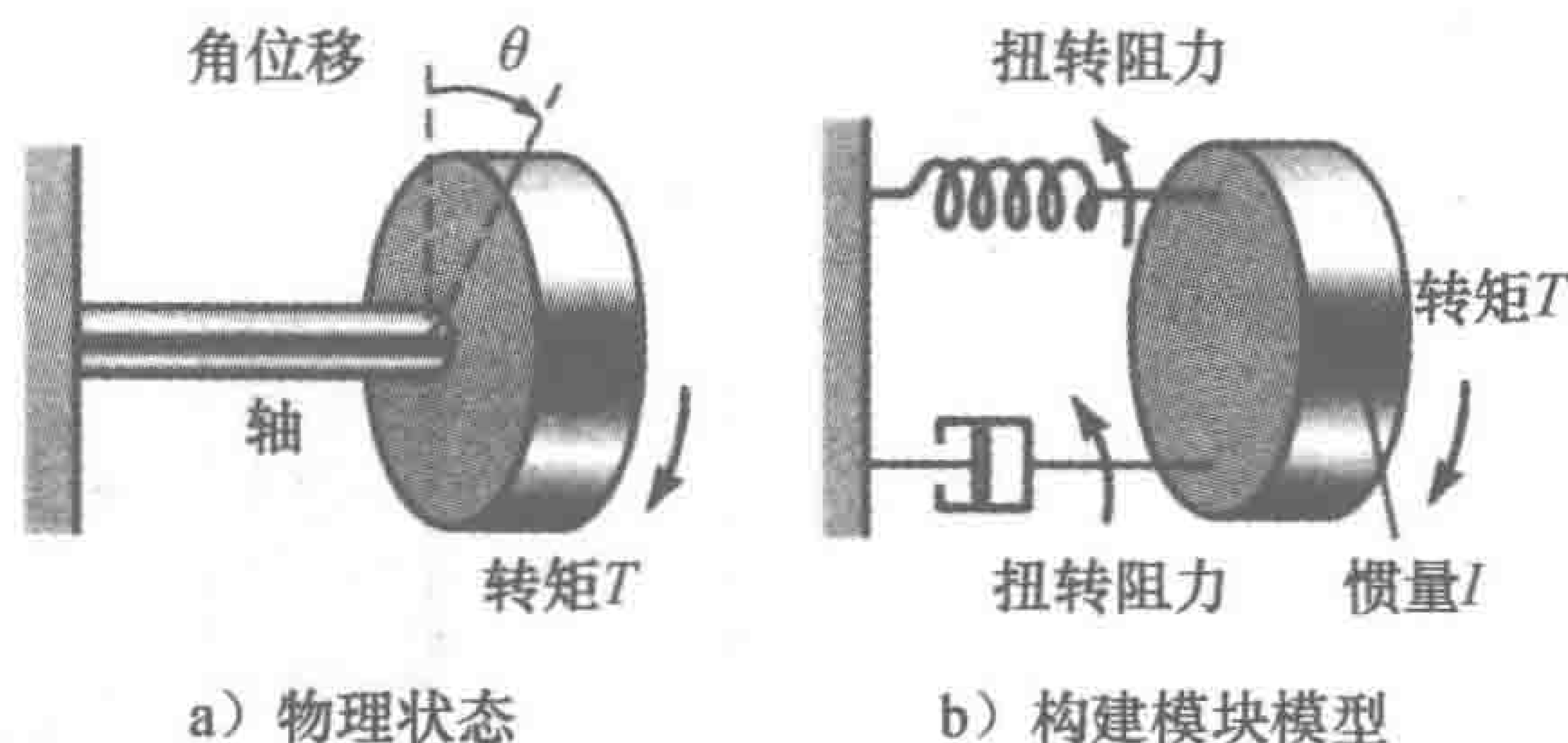


图 17-6　转动轴末端的质量块

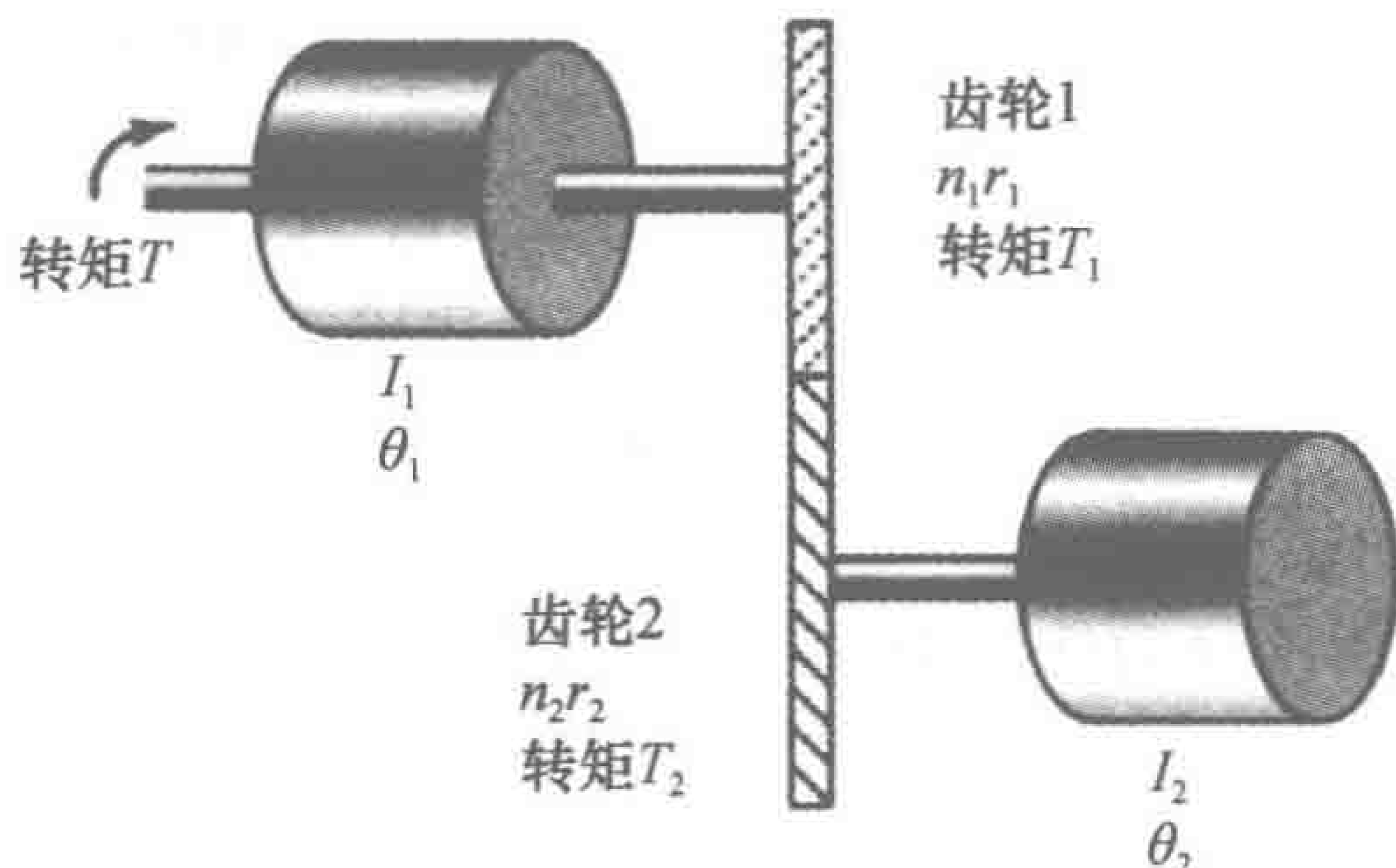


图 17-7　某两个齿轮传动的系统

如果齿轮 1 转动 θ_1 角度，齿轮 2 转动 θ_2 角度，有

$$r_1\theta_1 = r_2\theta_2$$

齿轮的半径比 n 等于齿轮的齿数比

$$\frac{r_1}{r_2} = \frac{n_1}{n_2} = n$$

如果给系统施加转矩 T，并给齿轮 1 施加一个转矩 T_1，其净转矩为 $T-T_1$，则有

$$T - T_1 = I_1 \frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2}$$

如果齿轮 2 的转矩为 T_2，有

$$T_2 = I_2 \frac{\mathrm{d}^2\theta_2}{\mathrm{d}t^2}$$

假设齿轮 1 传递的功率等于齿轮 2 传递的功率，因为传递功率为转矩和角速度的乘积，所以有

$$T_1 \frac{\mathrm{d}\theta_1}{\mathrm{d}t} = T_2 \frac{\mathrm{d}\theta_2}{\mathrm{d}t}$$

因为 $r_2\theta_1 = r_2\theta_2$，所以有

$$r_1 \frac{\mathrm{d}\theta_1}{\mathrm{d}t} = r_2 \frac{\mathrm{d}\theta_2}{\mathrm{d}t}$$

因此

$$\frac{T_1}{T_2} = \frac{r_1}{r_2} = n$$

420~421

则有

$$T - T_1 = T - nT_2 = T - n\left(I_2 \frac{\mathrm{d}^2\theta_2}{\mathrm{d}t^2}\right)$$

我们可以写成

$$T - n\left(I_2 \frac{\mathrm{d}^2\theta_2}{\mathrm{d}t^2}\right) = I_1 \frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2}$$

由于 $\theta_2 = n\theta_1$，$\mathrm{d}\theta_2/\mathrm{d}t = n\mathrm{d}\theta_1/\mathrm{d}t$，$\mathrm{d}^2\theta_2/\mathrm{d}t^2 = n\mathrm{d}^2\theta_1/\mathrm{d}t^2$，所以有

$$T - n^2\left(I_2 \frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2}\right) = I_1 \frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2}$$

$$(I_1 + n^2 I_2) \frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2} = T$$

不考虑齿轮系，可简化为：

$$I_1 \frac{\mathrm{d}^2\theta_1}{\mathrm{d}t^2} = T$$

因此，负载的转动惯量被以附加的转动惯量项 $n^2 I_2$ 折算到齿轮系的另一端。

17.3　电气系统模块

电气系统的基本模块包括电感、电容和电阻（见图 17-8）。

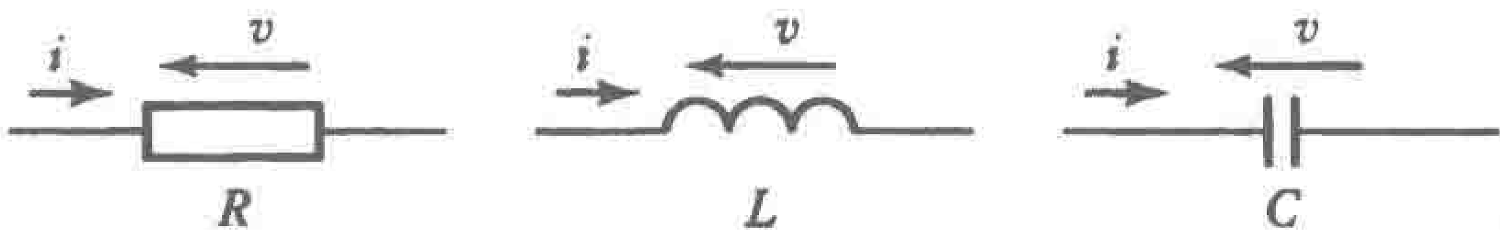


图 17-8　电气模块

在任何时刻**电感**上的感应电压 v 依赖于其流过电流的变化率 $\mathrm{d}i/\mathrm{d}t$：

$$v = L \frac{\mathrm{d}i}{\mathrm{d}t}$$

其中 L 是电感。感应电压的方向与通过电感的电流变化方向相反。电流方程为

$$i = \frac{1}{L}\int v \mathrm{d}t$$

对于**电容**，其电压与电容两端瞬间存储的电荷 q 有关：

422

$$v = \frac{q}{C}$$

其中 C 是电容。由于流入或者流出电容的电流 i 是进、出电容极板电荷的变化率，即 $i = \mathrm{d}q/\mathrm{d}t$，因此极板上总的电荷 q 为

$$q = \int i \mathrm{d}t$$

且

$$v = \frac{1}{C}\int i \mathrm{d}t$$

另外，因为 $v = q/C$，则

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{1}{C}\frac{\mathrm{d}q}{\mathrm{d}t} = \frac{1}{C}i$$

所以有

$$i = C\frac{\mathrm{d}v}{\mathrm{d}t}$$

对于**电阻**，其电压 v 与其瞬时电流 i 相关：

$$v = Ri$$

R 为电阻。

电容和电感能存储能量并进行释放，电阻只消耗能量。电感在电流 i 下存储的能量为

$$E = \frac{1}{2}Li^2$$

在电压为 v 时电容存储的能量为

$$E = \frac{1}{2}Cv^2$$

电阻在电压为 v 时消耗的功率为

$$P = iv = \frac{v^2}{R}$$

表 17-2 总结了在电流作为输入，电压作为输出时电气模块的方程。请与表 17-1 中的机械模块的方程进行比较。

表 17-2 电气模块

模块	描述方程	能量存储或功率消耗
电感	$i = \frac{1}{L}\int v \mathrm{d}t$	$E = \frac{1}{2}Li^2$
	$v = L\frac{\mathrm{d}i}{\mathrm{d}t}$	
电容	$i = C\frac{\mathrm{d}v}{\mathrm{d}t}$	$E = \frac{1}{2}Cv^2$
电阻	$i = \frac{v}{R}$	$P = \frac{v^2}{R}$

423

17.3.1 构建电气系统模型

描述电气模块如何组合的方程是**基尔霍夫定律**（Kirchhoff's laws）。该定律可表达为：

定律 1：流入一个节点的总电流等于流出该点的总电流，即该节点上电流的代数和为零。

定律 2：在一个闭合回路中，电路中每个部分电压的代数和等于输入的电压。

考虑图 17-9 中一个电阻和电容串联的简单电气系统，应用基尔霍夫第二定律有

$$v = v_R + v_C$$

其中 v_R 是电阻上的电压，v_C 是电容上的电压。由于这只是一个单回路，所以流过电路每一个元件的电流 i 相等。如果电路的输出为电容的电压 v_C，由于 $v_R = iR$ 以及 $i = C(\mathrm{d}v_C/\mathrm{d}t)$，所以有

$$v = RC\frac{\mathrm{d}v_C}{\mathrm{d}t} + v_C$$

从而得知输出 v_C 与输入 v 的关系为一个一阶微分方程。

图 17-10 为一个电阻－电感－电容系统。应用基尔霍夫第二定律，有

$$v = v_R + v_L + v_C$$

其中，v_R 是电阻上的电压，v_L 是电感上的电压，v_C 是电容上的电压。由于这只是一个单回路，所以流过电路每一个部分的电流 i 相同。如果电路的输出为电容的电压 v_C，由于 $v_R = iR$ 以及 $v_L = L(\mathrm{d}i/\mathrm{d}t)$，所以有

$$v = iR + L\frac{\mathrm{d}i}{\mathrm{d}t} + v_C$$

但 $i = C(\mathrm{d}v_C/\mathrm{d}t)$，则有

$$\frac{\mathrm{d}i}{\mathrm{d}t} = C\frac{\mathrm{d}(\mathrm{d}v_C/\mathrm{d}t)}{\mathrm{d}t} = C\frac{\mathrm{d}^2v_C}{\mathrm{d}t^2}$$

因此

$$v = RC\frac{\mathrm{d}v_C}{\mathrm{d}t} + LC\frac{\mathrm{d}^2v_C}{\mathrm{d}t^2} + v_C$$

这是一个二阶微分方程。

424

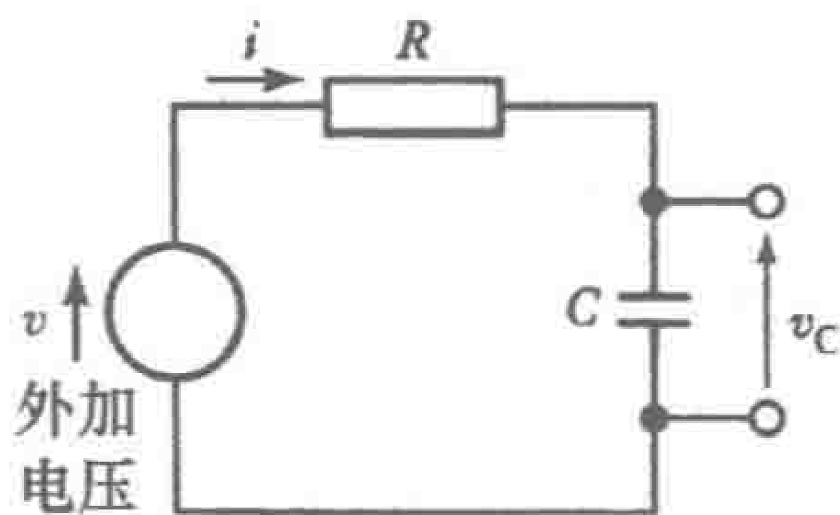


图 17-9　电阻－电容系统

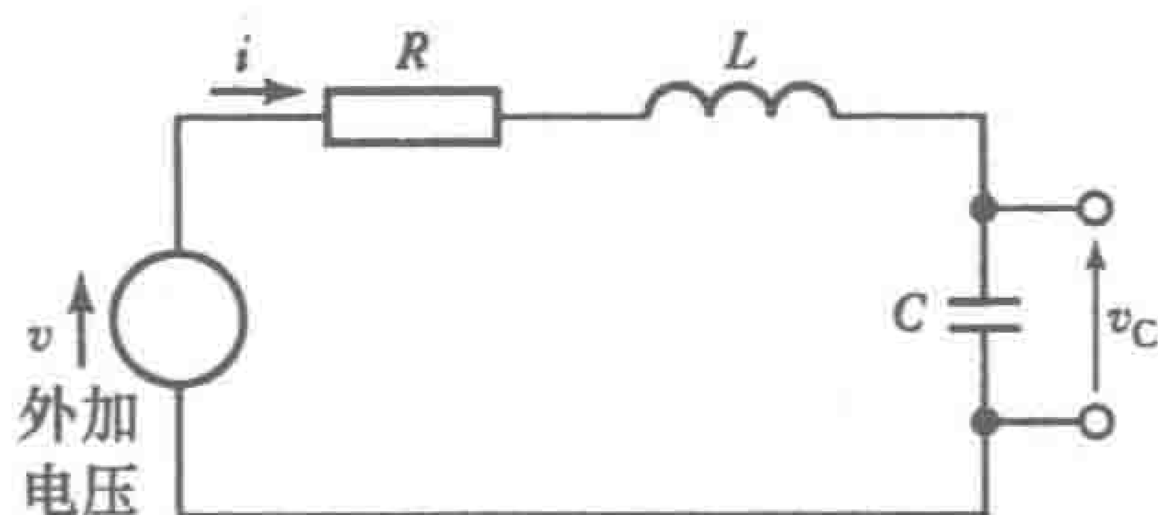


图 17-10　电阻－电感－电容系统

作为进一步的说明，考虑图 17-11 电路中电感的电压 v_L 为输出、v 为输入的二者之间的关系，应用基尔霍夫第二定律有

$$v = v_R + v_L$$

其中 v_R 是电阻上的电压，v_L 是电感上的电压。由于 $v_R = iR$，所以有

$$v = iR + v_L$$

由于

$$i = \frac{1}{L}\int v_L \mathrm{d}t$$

所以输入/输出之间的关系为

$$v = \frac{R}{L}\int v_L \mathrm{d}t + v_L$$

作为另外一个例子，考虑图 17-12 中电容两端电压 v_C 作为输出，电压 v 作为输入的二者之间的关系，应用基尔霍夫第一定律有

$$i_1 = i_2 + i_3$$

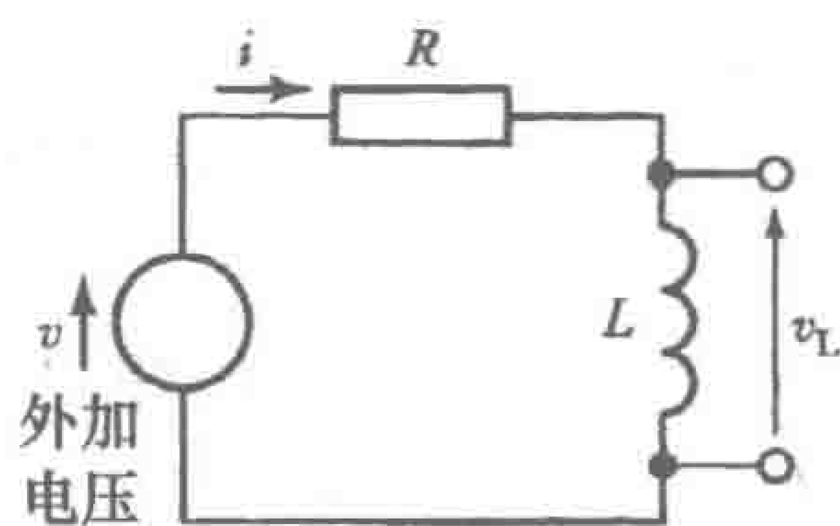


图 17-11　电阻 - 电感系统

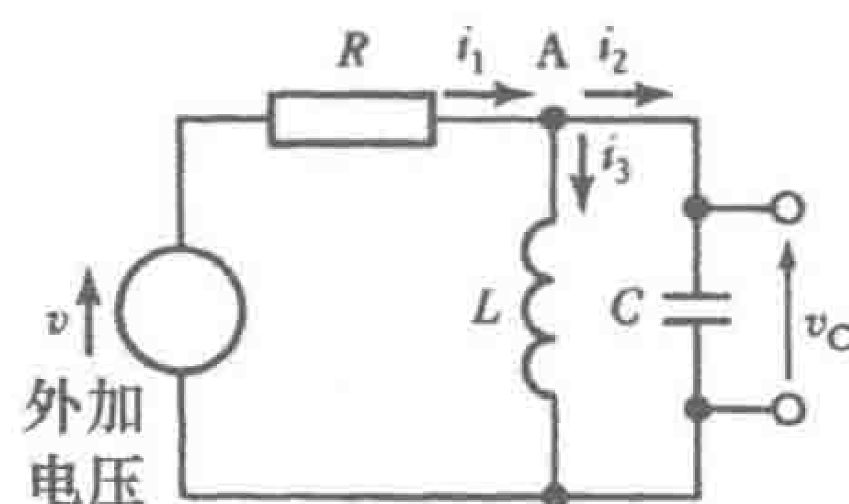


图 17-12　电阻 - 电容 - 电感系统

但

$$i_1 = \frac{v - v_A}{R}$$

$$i_3 = \frac{1}{L}\int v_A \mathrm{d}t$$

$$i_2 = C\frac{\mathrm{d}v_A}{\mathrm{d}t}$$

因此

$$\frac{v - v_A}{R} = \frac{1}{L}\int v_A \mathrm{d}t + C\frac{\mathrm{d}v_A}{\mathrm{d}t}$$

但 $v_C = v_A$。因此经变换有

$$v = RC\frac{dv_C}{\mathrm{d}t} + v_C + \frac{R}{L}\int v_C \mathrm{d}t$$

17.3.2　电气和机械系统的比较

电气和机械系统的基本模块具有很多类似之处（见图 17-13）。例如，电路中电阻不能存储能量，但能消耗能量，当电流 i 流过电阻时有 $i = v/R$，其中 R 是一个常量，功率 P 为 $P = v^2/R$。机械系统中类似于电阻的基本模块为阻尼器。它同样不存储能量，但消耗能量，在力 F 与速度 v 的关系为 $F = cv$，其中 c 是常量，功率 P 为 $P = cv^2$。这些方程具有类似的形式。通过比较，把电流类比成力，电压类比成速度，阻尼常量 c 类比为电阻的倒数，即 $(1/R)$。通过电流和力、电压和速度的类比，可以将弹簧类比为电感，质量类比为电容。

425

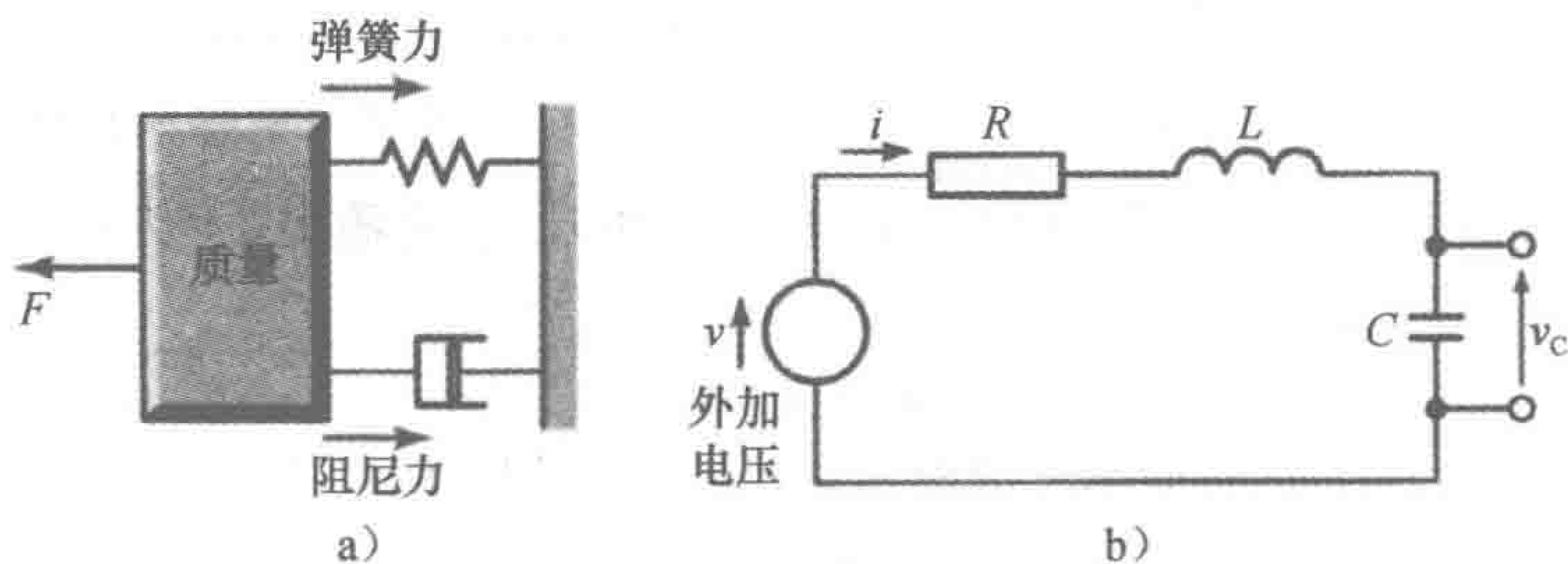


图 17-13　模拟系统

图 17-13a 中的机械系统和图 17-13b 的电气系统具有类似的输入/输出微分方程：

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + c\frac{\mathrm{d}x}{\mathrm{d}t} + \mathrm{k}x = F \text{ 和 } RC\frac{\mathrm{d}v_C}{\mathrm{d}t} + LC\frac{\mathrm{d}^2v_C}{\mathrm{d}t^2} + v_C = v$$

电流和力的类比是最常用的。然而，有些情况下也可以将电压与力进行类比。

17.4　流体系统模块

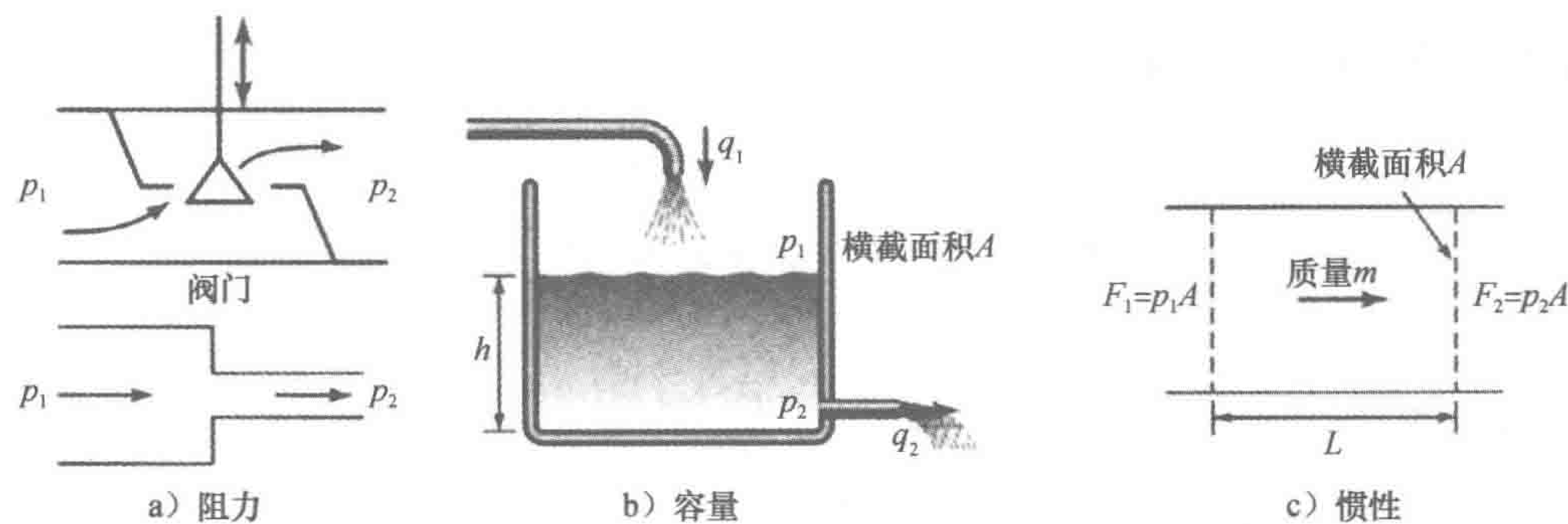
在流体系统中有三个基本模块，可等效为电路中的电阻、电容和电感。流体系统可以简化

为两类：一类是液压系统，其中流体是液体，并认为是不能压缩的；另一类是气压系统，其中气体能被压缩并改变密度。

液阻是流体在经过阀门或者管径改变处产生的流动受到阻碍的现象（见图 17-14a）。流过阻尼部件时，流体流量 q 和压差（p_1-p_2）的关系是

$$p_1 - p_2 = Rq$$

其中 R 是液阻常量。阻值越大，给定流速下压力差越大。这个方程与电路中欧姆定理相似，假设为线性关系。当液体规则流过毛细管或多孔塞时液阻为线性的，而当液体流经尖锐的边界口或发生湍流时则为非线性的。



a）阻力　　b）容量　　c）惯性

图 17-14　液压示例

液容是用来描述以势能的形式存储于液体中的能量。液体在容器中的高度（见图 17-14b），或称为压力水头（pressure head），是能量存储的一种形式。对于这种存储，容器中容量 V 的变化率 $\mathrm{d}V/\mathrm{d}t$，等于在液体进入容器时的流量 q_1 与流出时的流量 q_2 之差：

$$q_1 - q_2 = \frac{\mathrm{d}V}{\mathrm{d}t}$$

但 $V=Ah$，其中 A 是容器的横截面积，h 为液体在容器中的高度。因此

$$q_1 - q_2 = \frac{\mathrm{d}(Ah)}{\mathrm{d}t} = A\frac{\mathrm{d}h}{\mathrm{d}t}$$

但输入与输出之间的压差为 p，其中 $p=h\rho g$，ρ 是液体的密度，g 是重力加速度。因此，如果液体是不能压缩的，即在压力下液体的密度不会改变，则

$$q_1 - q_2 = A\frac{\mathrm{d}(p/\rho g)}{\mathrm{d}t} = \frac{A}{\rho g}\frac{\mathrm{d}p}{\mathrm{d}t}$$

液容 C 定义为

$$C = \frac{A}{\rho g}$$

因此

$$q_1 - q_2 = C\frac{\mathrm{d}p}{\mathrm{d}t}$$

以上方程求积分，可得

$$p = \frac{1}{C}\int(q_1 - q_2)\,\mathrm{d}t$$

液感等价于电气系统中的电感或者机械系统中的弹簧。若需要加快流速，则需要提供力。考虑液体质量模块 m（见图 17-14c），施加在液体上的净力为：

$$F_1 - F_2 = p_1A - p_2A = (p_1 - p_2)A$$

426 ~ 427

其中压差为（p_1-p_2），A 是截面积。净力导致质量的加速度 a，则有

$$(p_1-p_2)A = ma$$

加速度 a 是速度的变化率 $\mathrm{d}v/\mathrm{d}t$，则有

$$(p_1-p_2)A = m\frac{\mathrm{d}v}{\mathrm{d}t}$$

假设液体的体积为 AL，其中 L 是液体的长度，或液压 p_1 和 p_2 测量处的距离。如果液体的密度为 ρ，有 $m=AL\rho$，因此

$$(p_1-p_2)A = AL\rho\frac{\mathrm{d}v}{\mathrm{d}t}$$

体积的变化率为 $q=Av$，从而

$$(p_1-p_2)A = L\rho\frac{\mathrm{d}q}{\mathrm{d}t}$$

$$p_1-p_2 = I\frac{\mathrm{d}q}{\mathrm{d}t}$$

液感 I 的定义为

$$I = \frac{L\rho}{A}$$

在气动系统中这三个基本模块与液压系统类似，为气阻、气容和气感。气体与液体不同的是，气体是可压缩的，压力的变化将导致容量和密度的变化。**气阻** R 的定义是依据质量的流动率 $\mathrm{d}m/\mathrm{d}t$（通常此处在 m 上加一点表示流体的质量流动率而不表示质量）与压差（p_1-p_2）：

$$p_1-p_2 = R\frac{\mathrm{d}m}{\mathrm{d}t} = R\dot{m}$$

由于气体的压缩特性，**气容** C 类似于弹簧的压缩并存储能量。如果气体以质量流动速率 $\mathrm{d}m_1/\mathrm{d}t$ 进入容量为 V 的容器，并以质量流动速率 $\mathrm{d}m_2/\mathrm{d}t$ 流出容器，从而容器中质量的变化率为（$\mathrm{d}m_1/\mathrm{d}t-\mathrm{d}m_2/\mathrm{d}t$）。如果气体在容器中的密度为 ρ，则容器中气体质量的变化率为

$$\text{容器中质量的变化率} = \frac{\mathrm{d}(\rho V)}{\mathrm{d}t}$$

由于气体是可压缩的，ρ 和 V 都随时间变化，所以

428

$$\text{容器中质量的变化率} = \rho\frac{\mathrm{d}V}{\mathrm{d}t} + V\frac{\mathrm{d}\rho}{\mathrm{d}t}$$

由于$(\mathrm{d}V/\mathrm{d}t)=(\mathrm{d}V/\mathrm{d}p)(\mathrm{d}p/\mathrm{d}t)$，对于理想气体，有 $pV=mrT$，$p=(m/V)rT=\rho rT$ 以及 $\mathrm{d}\rho/\mathrm{d}t=(1/rT)(\mathrm{d}p/\mathrm{d}t)$，所以

$$\text{容器中质量的变化率} = \rho\frac{\mathrm{d}V}{\mathrm{d}p}\frac{\mathrm{d}p}{\mathrm{d}t} + \frac{V}{rT}\frac{\mathrm{d}p}{\mathrm{d}t}$$

其中 r 是气体常量，T 是温度，假设也是一个开氏（Kelvin）温度下的常量，有

$$\frac{\mathrm{d}m_1}{\mathrm{d}t} - \frac{\mathrm{d}m_2}{\mathrm{d}t} = \left(\rho\frac{\mathrm{d}V}{\mathrm{d}p} + \frac{V}{rT}\right)\frac{\mathrm{d}p}{\mathrm{d}t}$$

由容器体积变化引起的气容 C_1 定义为

$$C_1 = \rho\frac{\mathrm{d}V}{\mathrm{d}p}$$

由气体的压缩而引起的气容 C_2 为

$$C_2 = \frac{V}{rT}$$

因此

$$\frac{\mathrm{d}m_1}{\mathrm{d}t}-\frac{\mathrm{d}m_2}{\mathrm{d}t}=(C_1+C_2)\frac{\mathrm{d}p}{\mathrm{d}t}$$

或者

$$p_1-p_2=\frac{1}{C_1+C_2}\int(\dot{m}_1-\dot{m}_2)\mathrm{d}t$$

气感是指让气体产生加速度所需要的气压降。根据牛顿第二定理，净力为 $ma=\mathrm{d}(mv)/\mathrm{d}t$。而力是由压力差（$p_1-p_2$）提供，若 A 为加速气体块的横截面积，则

$$(p_1-p_2)A=\frac{\mathrm{d}(mv)}{\mathrm{d}t}$$

加速气体的质量 m 等于 ρLA，其中 ρ 为气体密度，L 是加速气体块的长度。气体体积的流量为 $q=Av$，其中 v 是速度。从而

$$mv=\rho LA\frac{q}{A}=\rho Lq$$

和

$$(p_1-p_2)A=L\frac{\mathrm{d}(\rho q)}{\mathrm{d}t}$$

但 $\dot{m}=\rho q$，因此

$$p_1-p_2=\frac{L}{A}\frac{\mathrm{d}\dot{m}}{\mathrm{d}t}$$

$$p_1-p_2=I\frac{\mathrm{d}\dot{m}}{\mathrm{d}t}$$

其中气感 I 为 L/A。

429

表 17-3 列出了流体系统基本模块的基本特性，包括液压和气动系统。

表 17-3　液压及气压模块

模块	描述方程	能量存储或功率消耗
液体		
流感	$q=\frac{1}{I}\int(p_1-p_2)\mathrm{d}t$	$E=\frac{1}{2}Iq^2$
	$p=I\frac{\mathrm{d}q}{\mathrm{d}t}$	
液容	$q=C\frac{\mathrm{d}(p_1-p_2)}{\mathrm{d}t}$	$E=\frac{1}{2}C(p_1-p_2)^2$
液阻	$q=\frac{p_1-p_2}{R}$	$p=\frac{1}{R}(p_1-p_2)^2$
气体		
气感	$\dot{m}=\frac{1}{I}\int(p_1-p_2)\mathrm{d}t$	$E=\frac{1}{2}I\dot{m}^2$
气容	$\dot{m}=C\frac{\mathrm{d}(p_1-p_2)}{\mathrm{d}t}$	$E=\frac{1}{2}C(p_1-p_2)^2$
气阻	$\dot{m}=\frac{p_1-p_2}{R}$	$p=\frac{1}{R}(p_1-p_2)^2$

对于液压系统流体的容积变化率和气动系统流体的质量变化率是与电气系统中的电流对应的。液压和气动系统中的压差对应于电气系统中的电压。对比表 17-3 和表 17-2，液压和气动的容抗和感抗是储能环节，而液压和气动的阻抗消耗能量。

17.4.1　构建流体系统模型

图 17-15 中是一个简单的液压系统，液体流入并流出容器。这个系统可认为是由一个液容

（指盛放液体的容器）和液阻（指阀门）组成的。

若液体的流速很慢，则感抗可以忽略。对于液容，我们有

$$q_1 - q_2 = C\frac{\mathrm{d}p}{\mathrm{d}t}$$

液体流出容器的速度 q_2 等于液体流出阀门的速度。从而对于液阻有

$$p_1 - p_2 = Rq_2$$

压差（$p_1 - p_2$）是由液体在容器中的高度引起的，为 $h\rho g$。从而 $q_2 = h\rho g/R$，代替第一个式中的 q_2，有

$$q_1 - \frac{h\rho g}{R} = C\frac{\mathrm{d}(h\rho g)}{\mathrm{d}t}$$

由于 $C = A/\rho g$，所以

$$q_1 = A\frac{\mathrm{d}h}{\mathrm{d}t} + \frac{\rho gh}{R}$$

这个方程描述了容器中液体的高度与输入液体的流量之间的关系。

风箱是一个简单的气动系统例子（见图 17-16）。入口的缩颈限制了气体进入风箱的速度，使系统存在气阻，风箱本身提供了气容。在流速变化慢的情况下，感抗可以忽略。

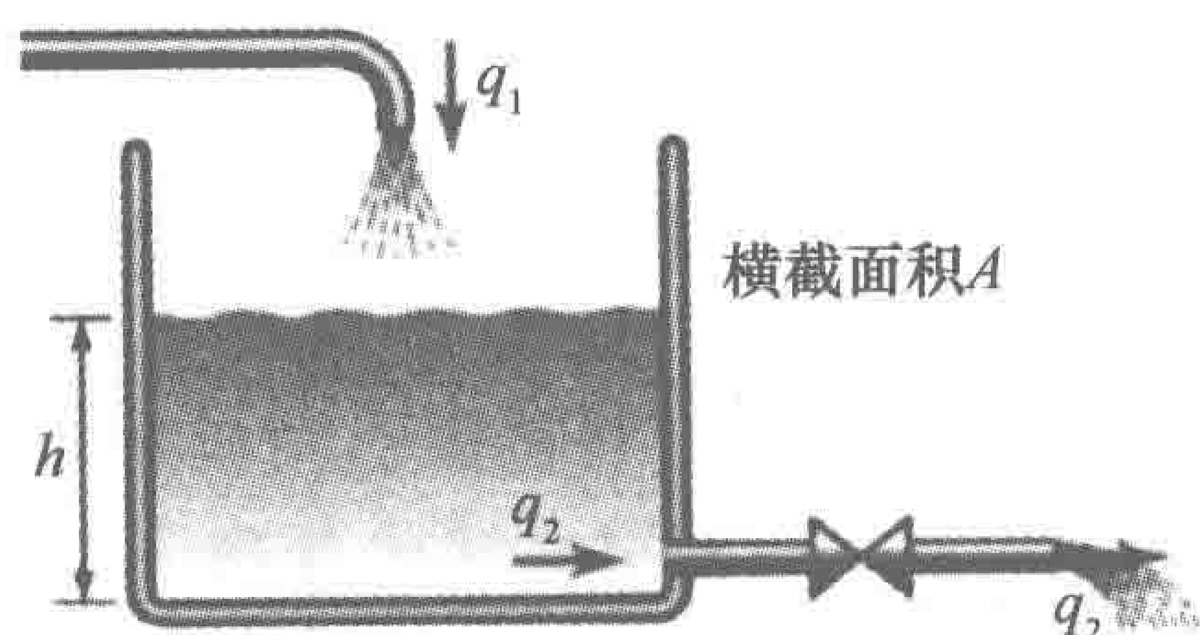


图 17-15　一个流体系统

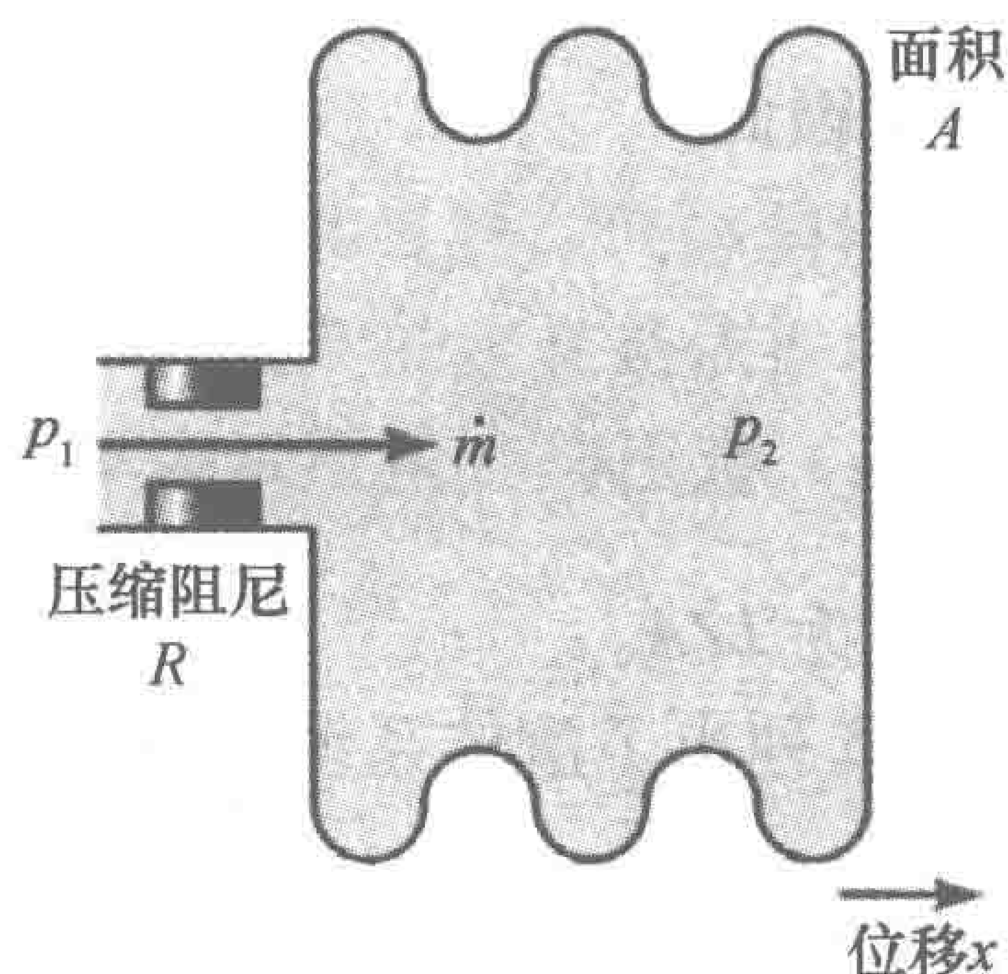


图 17-16　一个气动系统

质量流速为

$$p_1 - p_2 = R\dot{m}$$

其中 p_1 是缩颈之前的压力，p_2 是缩颈之后的压力（即风箱中的气压）。因为没有出口，所以气体流入风箱后被留在风箱中而不会流出。风箱的气容由下式给出：

$$\dot{m}_1 - \dot{m}_2 = (C_1 + C_2)\frac{\mathrm{d}p_2}{\mathrm{d}t}$$

流入风箱的质量流速由气阻方程给出，且流出的质量为零。故

$$\frac{p_1 - p_2}{R} = (C_1 + C_2)\frac{\mathrm{d}p_2}{\mathrm{d}t}$$

因此

$$p_1 = R(C_1 + C_2)\frac{\mathrm{d}p_2}{\mathrm{d}t} + p_2$$

这个方程描述了在输入气压为 p_1 时，风箱中的气压 p_2 与时间的变化关系。

430～431

风箱内气压的变化将导致风箱膨胀或者收缩。可以把风箱当成弹簧，导致风箱伸缩的力 F 与位移 x 之间的关系可以写成 $F = kx$，其中 k 是风箱的弹簧常量。而压力 p_2 决定了力 F，其中 $p_2 = F/A$，A 是风箱的横截面积，因此 $p_2A = F = kx$。把 p_2 代入前面的方程有

$$p_1 = R(C_1 + C_2)\frac{k}{A}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{k}{A}x$$

这是一个一阶微分方程，表示在输入压力为 p_1 时风箱膨胀或者收缩的长度 x 与时间的关系。由容器体积变化引起的气容 C_1 为 $\rho dV/dp_2$，因为 $V=Ax$，所以 $C_1=\rho A dx/dp_2$。对于风箱有 $p_2A=kx$，故

$$C_1 = \rho A \frac{dx}{d(kx/A)} = \frac{\rho A^2}{k}$$

由于气体压缩而导致的气容 C_2 为 $V/rT=Ax/rT$。

下面分析图 17-17 中的液压系统，推导两个容器中液体高度随时间的变化关系。在这个模型中忽略感抗。

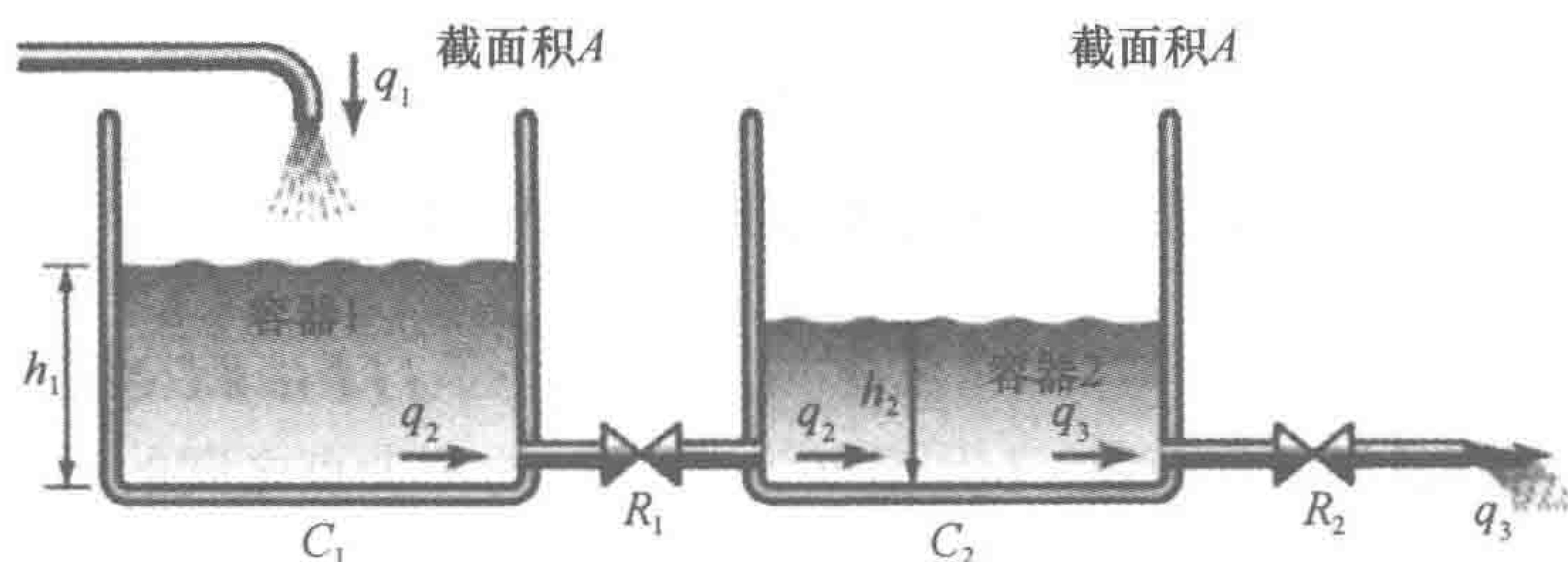


图 17-17　一个流体系统

容器 1 为一个液容，故有

$$q_1 - q_2 = C_1 \frac{dp}{dt}$$

其中 $p=h_1\rho g$ 且 $C_1=A_1/\rho g$，因此有

$$q_1 - q_2 = A_1 \frac{dh_1}{dt}$$

容器中液体流出的流量 q_2 等于阀门流出的流量，对于阀门的液阻 R_1，有

$$p_1 - p_2 = R_1 q_2$$

压力为 $h_1\rho g$ 和 $h_2\rho g$。因此

$$(h_1 - h_2)\rho g = R_1 q_2$$

将这个方程中 q_2 的值，代入上一个方程，有

$$q_1 - \frac{(h_1 - h_2)\rho g}{R_1} = A_1 \frac{dh_1}{dt}$$

432

这个方程为容器 1 中液体的高度与输入流量之间的关系。

容器 2 有类似的推导过程，对于液容 C_2，有

$$q_2 - q_3 = C_2 \frac{dp}{dt}$$

其中 $p=h_2\rho g$ 且 $C_2=A_2/\rho g$，因此

$$q_2 - q_3 = A_2 \frac{dh_2}{dt}$$

液体流出容器的流量 q_3 等于在阀门 R_2 流出的流量，对于阀门的液阻，有

$$p_2 - 0 = R_2 q_3$$

假设液体流到外部大气中，将本式中 q_3 代入前一个式子，有

$$q_2 - \frac{h_2\rho g}{R_2} = A_2 \frac{dh_2}{dt}$$

将 q_2 的值由第一个容器所计算的值替代，有

$$\frac{(h_1 - h_2)\rho g}{R_1} - \frac{h_2\rho g}{R_2} = A_2 \frac{dh_2}{dt}$$

这个方程描述了容器 2 中液体高度随时间的变化关系。

17.5　热力系统模块

热力系统只有两种模块：热阻和热容。如果两点之间的温度不同，就会产生热流。这种情况的等效电路就是两点之间如果存在电压 v，就会产生电流 i，电流与电压之间的关系是 $i = v/R$，其中，R 是两点之间的电阻。可以用类似的关系来定义**热阻** R。如果 q 是热流的流量，$(T_1 - T_2)$ 是温度差，则

$$q = \frac{T_1 - T_2}{R}$$

热阻值由热传递的模式决定。在固体传导的情况下，对于单向传导：

$$q = Ak\frac{T_1 - T_2}{L}$$

433 其中，A 是热传导方向材料的截面积，L 是温度为 T_1、T_2 的两个点之间的材料长度。k 是导热系数，因此在这种热传导的模式中，

$$R = \frac{L}{Ak}$$

当热传导是对流时，就像液体和气体，则

$$q = Ah(T_1 - T_2)$$

其中 A 是存在温差的表面面积，h 是传热系数。因此，对于这种模式，有

$$R = \frac{1}{Ah}$$

热容是一个系统内部能量存储的度量。因此，如果热流入系统的流量为 q_1，流出流量为 q_2，那么

$$\text{内部能量变化速率} = q_1 - q_2$$

系统内部能量的增加意味着温度的提高。由于

$$\text{内部能量的变化} = mc \times \text{温度改变量}$$

其中，m 是质量，c 是指定的热容量，则，

$$\text{内部能量变化速率} = mc \times \text{温度变化速率}$$

因此

$$q_1 - q_2 = mc\frac{\mathrm{d}T}{\mathrm{d}t}$$

$\mathrm{d}T/\mathrm{d}t$ 是温度的变化速率。此方程可写为

$$q_1 - q_2 = C\frac{\mathrm{d}T}{\mathrm{d}t}$$

其中，C 是热容，因此 $C = mc$。表 17-4 总结了热系统的基本模块。

表 17-4　热系统模块

模块	描述方程	能量存储
热容	$q_1 - q_2 = C\frac{\mathrm{d}T}{\mathrm{d}t}$	$E = CT$
热阻	$q = \frac{T_1 - T_2}{R}$	

17.5.1　构建热系统模型

假设一个刚放入温度为 T_L 的液体的温度为 T 的温度计（见图 17-18）。 434

如果从液体到温度计的热流的热阻为 R，则

$$q = \frac{T_L - T}{R}$$

其中，q 是液体到温度计的净热流量。温度计的热容 C 由下式给出，

$$q_1 - q_2 = C\frac{\mathrm{d}T}{\mathrm{d}t}$$

由于只有从液体到温度计的净热流，$q_1 = q$ 且 $q_2 = 0$，所以有

$$q = C\frac{\mathrm{d}T}{\mathrm{d}t}$$

将本式中 q 代入上一个方程，有

$$C\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{T_L - T}{R}$$

重新整理得

$$RC\frac{\mathrm{d}T}{\mathrm{d}T} + T = T_L$$

这个方程为一阶微分方程，它反映了温度计插入热的液体中后其读数 T 随时间变化的关系。

假设上面的热系统中参数是集中的。例如系统中只有一个温度是针对温度计的，只有一个温度是针对液体的，即温度只是时间的函数，而与物体内的位置没有关系。

为了说明这一点，考虑图 17-19 中的热系统，由一个房间中的一个电热器组成。电热器的热能输出速率为 q_1，房子热能流失速率为 q_2。假设房间中空气的温度均为 T，而且墙壁不能吸收热量，下面推导房间温度是如何随时间变化的方程。

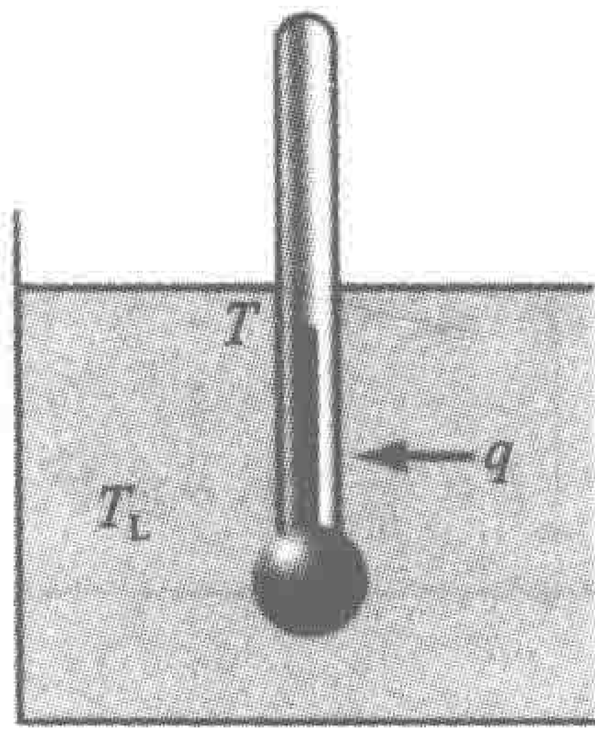


图 17-18　一个热系统

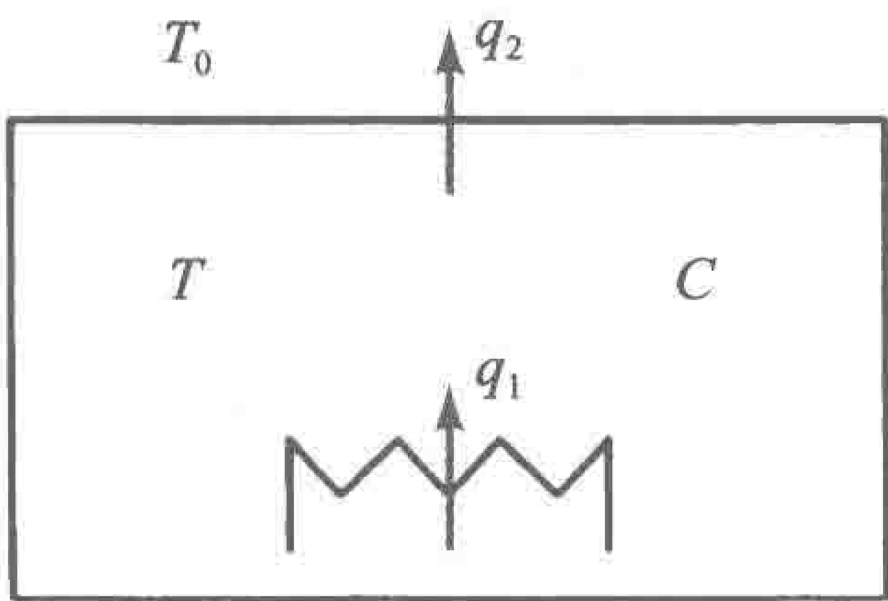


图 17-19　热系统

如果房间中空气的热容为 C，则

$$q_1 - q_2 = C\frac{\mathrm{d}T}{\mathrm{d}t}$$

如果定义房间内部的温度为 T，房间外的温度为 T_0，则

$$q_2 = \frac{T - T_0}{R}$$

其中 R 是墙壁的热阻。将 q_2 代入上式，有

$$q_1 - \frac{T - T_0}{R} = C\frac{\mathrm{d}T}{\mathrm{d}t}$$

因此

$$RC\frac{\mathrm{d}T}{\mathrm{d}t} + T = Rq_1 + T_0$$

435

本章小结

系统的**数学模型**是输入/输出关系方程的描述，因此根据输入可对输出进行预测。

机械系统可以简化为质量、弹簧和阻尼器三种基本模块，或者在转动情况下为转动惯量、弹簧和旋转阻尼器模块。电气系统可以简化为电阻、电容和电感模块，液压和气动系统为阻抗、容抗和感抗模块。热系统为热阻和热容模块。

在机械、电气、流体以及热系统中的很多模块具有类似的性质，例如机械系统中的质量与电气系统中的电容、流体系统中的液容（或气容）、热系统中的热容类似。表 17-5 中列出了不同系统中这些模块的比较以及它们的定义方程。

表 17-5 系统元件

	机械（平移）	机械（旋转）	电气	流体（液压）	热
元件	质量	转动惯量	电容	液容	热容
方程	$F=m\frac{d^2x}{dt^2}$	$T=I\frac{d^2\theta}{dt^2}$			
	$F=m\frac{dv}{dt}$	$T=I\frac{d\omega}{dt}$	$i=C\frac{dv}{dt}$	$q=C\frac{d(p_1-p_2)}{dt}$	$q_1-q_2=C\frac{dT}{dt}$
能量	$E=\frac{1}{2}mv^2$	$E=\frac{1}{2}I\omega^2$	$E=\frac{1}{2}Cv^2$	$E=\frac{1}{2}C(p_1-p_2)^2$	$E=CT$
元件	弹簧	弹簧	电感	液感	无
方程	$F=kx$	$T=k\theta$	$v=L\frac{di}{dt}$	$p=I\frac{dq}{dt}$	
能量	$E=\frac{1}{2}\frac{F^2}{k}$	$E=\frac{1}{2}\frac{T^2}{k}$	$E=\frac{1}{2}Li^2$	$E=\frac{1}{2}Iq^2$	
元件	阻尼器	旋转阻尼器	电阻	液阻	热阻
方程	$F=c\frac{d_x}{dt}=cv$	$T=c\frac{d\theta}{dt}=c\omega$	$i=\frac{v}{R}$	$q=\frac{p_1-p_2}{R}$	$q=\frac{T_1-T_2}{R}$
功率	$P=cv^2$	$P=c\omega^2$	$P=\frac{v^2}{R}$	$P=\frac{1}{R}(p_1-p_2)^2$	

436

习题

17.1 推导图 17-20 中描述的机械系统中输入力 F 与输出位移 x 的方程。

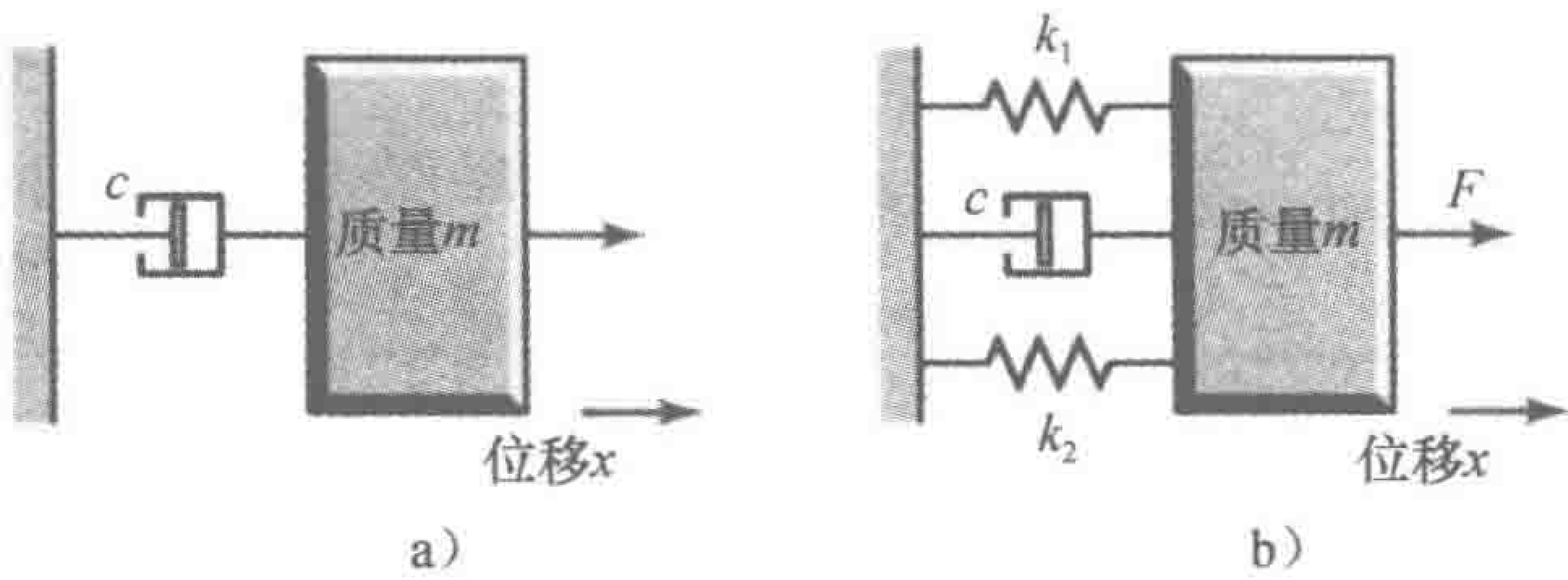


图 17-20 习题 17.1

17.2 针对在金属铁轨上运行的火车车厢的金属车轮，提出其模型。

17.3 推导图 17-21 中旋转系统的输入角位移 θ_i 和输出角位移 θ_o 之间的关系方程。

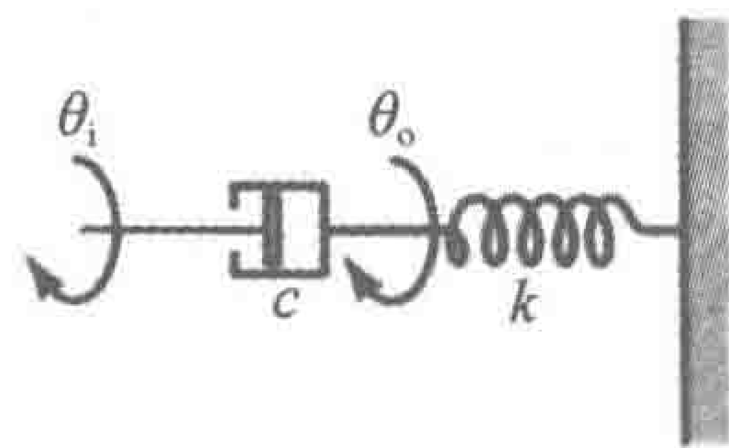


图 17-21　习题 17.3

17.4　对一个直径阶梯变化的阶梯轴带动质量块转动的系统，提出其等价模型，并推导输入转矩与输出转角的方程。可以忽略系统阻尼。

17.5　图 17-22 电路系统中为电阻与电容串联，输入为 v，输出为电阻 R 上的电压 v_R，求输入/输出的关系。

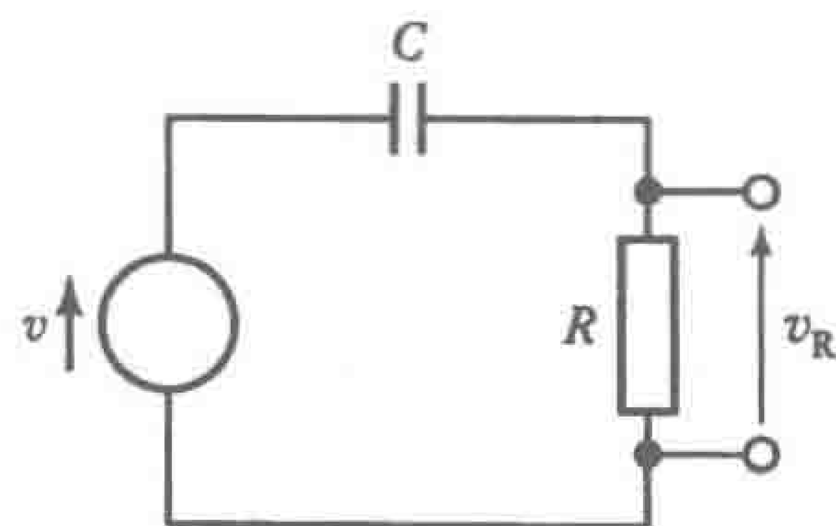


图 17-22　习题 17.5

17.6　推导图 17-23 中串联 LCR 电路的输入/输出关系，其中输入为 v，输出为电阻 R 两端电压 v_R。

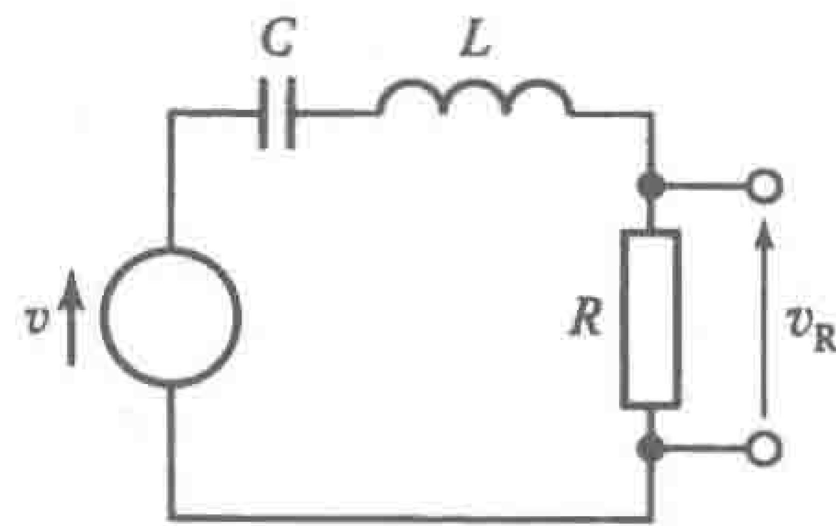


图 17-23　习题 17.6

17.7　推导图 17-24 中电路系统的输入 v 与输出电容 C 两端电压 v_C 之间的关系。

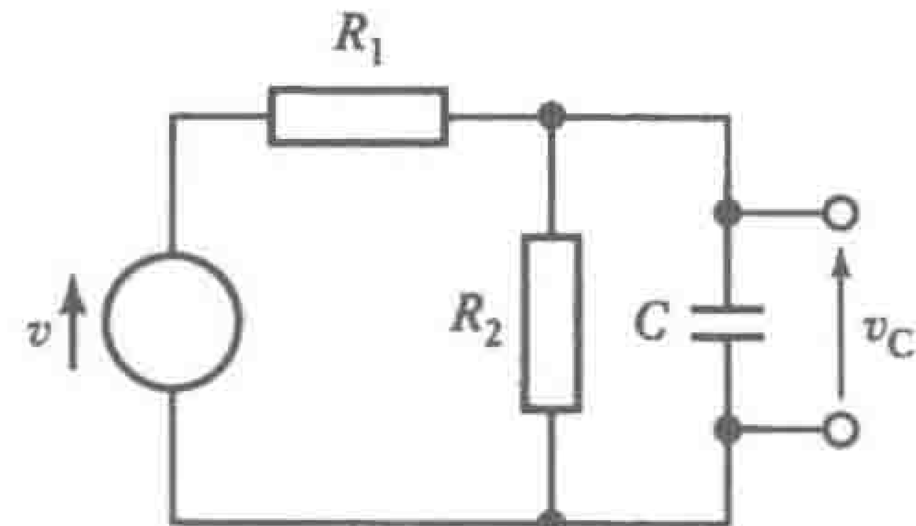


图 17-24　习题 17.7

17.8　推导图 17-25 中液压系统中液位高度 h_2 随时间的变化关系，忽略液感。

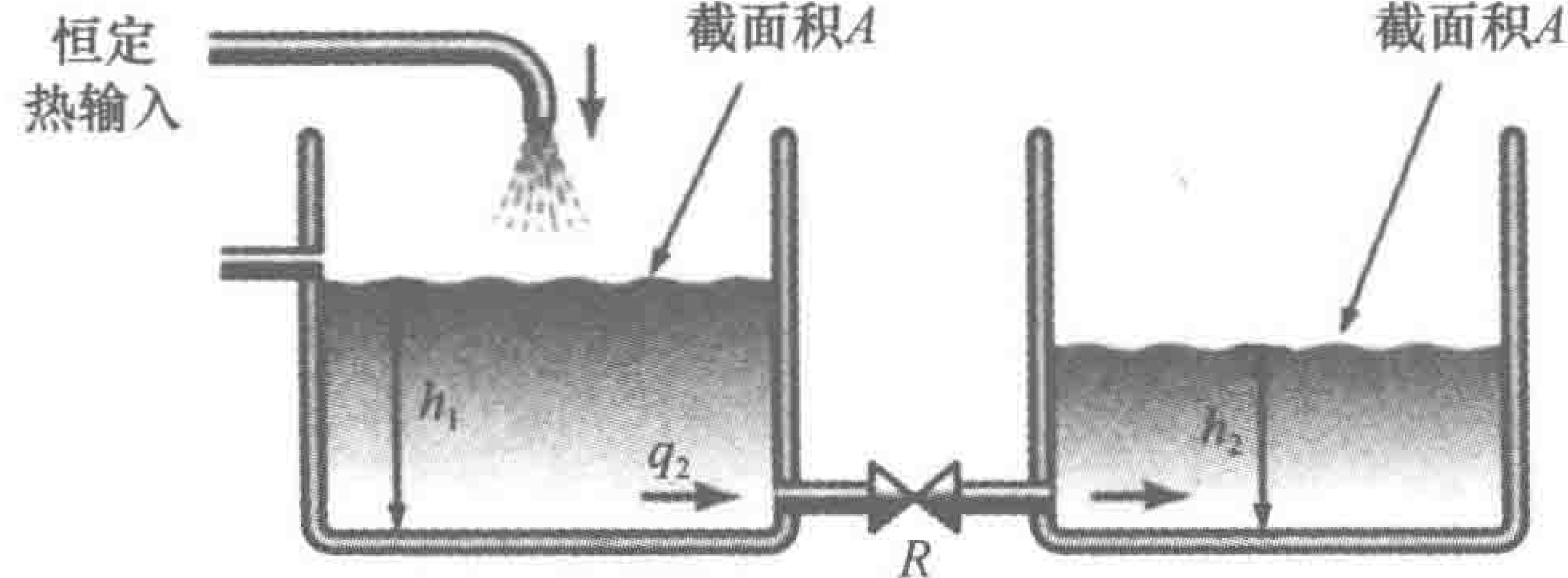


图 17-25　习题 17.8

17.9 一个热物体的热容为 C，温度为 T，放在温度为 T_r 的足够大的房间中冷却。假设热系统的热阻为 R，推导该物体的温度随时间变化的方程，给出等效的电气系统模型。

17.10 图 17-26 中的热系统中有两个隔离容器，其中一个具有加热器。如果具有加热器的部分的温度为 T_1，另外一部分的温度为 T_2，外部环境的温度为 T_3，推导 T_1 与 T_2 随时间的变化方程。其中，所有的容器壁具有相同热阻，并忽略热容。两个容器的热容均为 C。

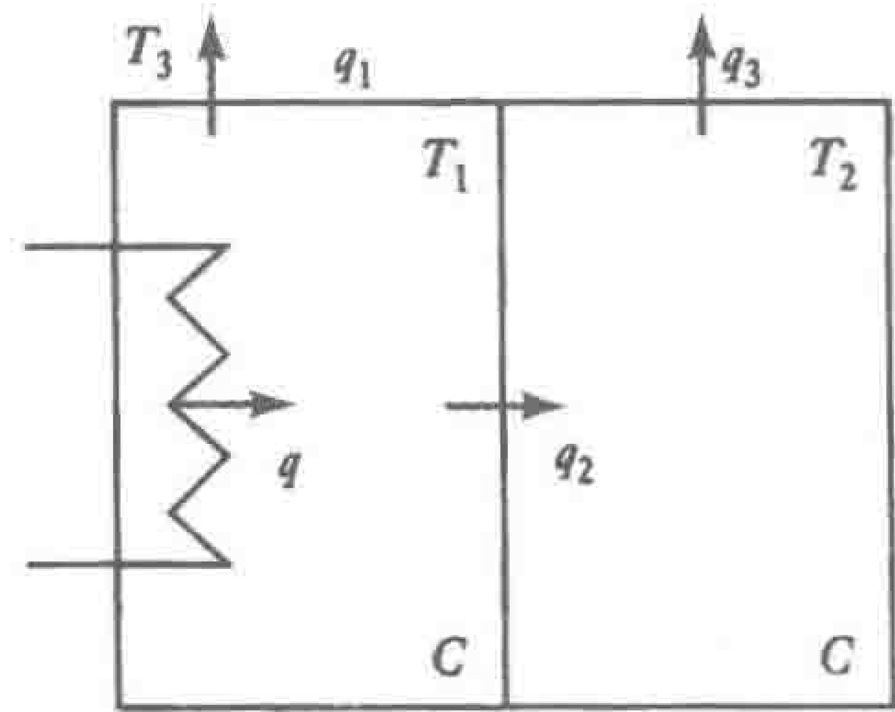


图 17-26 习题 17.10

17.11 推导关于图 7-23 薄膜驱动器输入压力 p 与杆子输出位移 x 之间的微分方程。

17.12 图 17-27 为电动机通过齿轮机构带动负载转动的机械系统，推导负载转动角度随时间变化的微分方程。

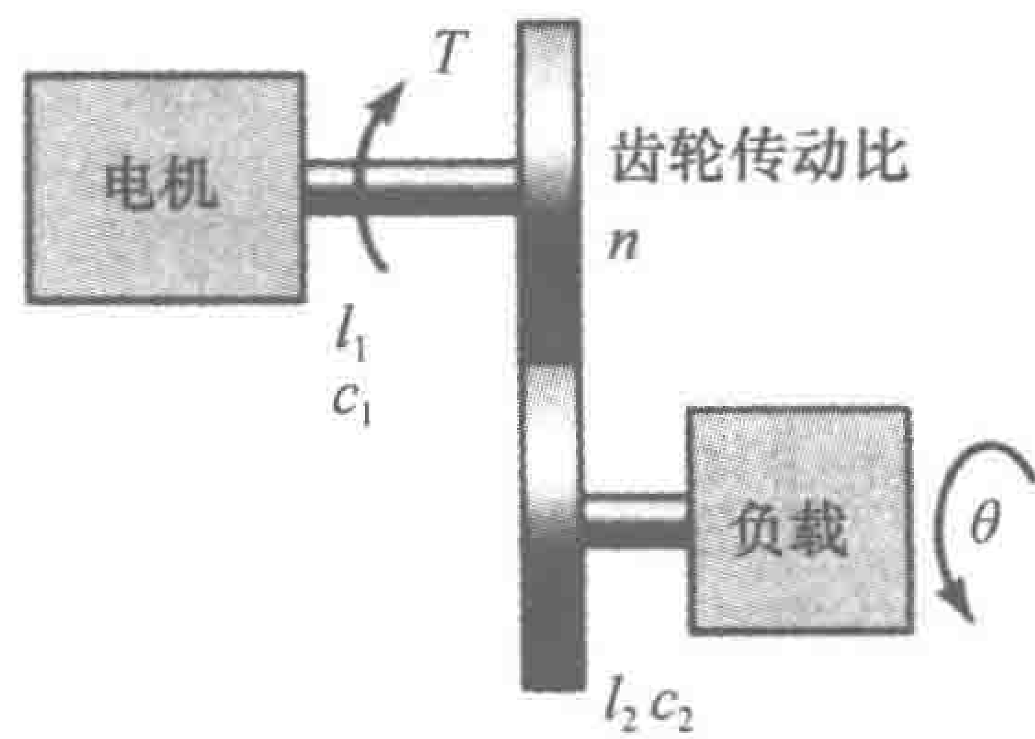


图 17-27 习题 17.12

第 18 章　工程系统模型

本章目的

通过本章的学习，读者将能够：

- 进行转动 - 平动、机电和液压 - 机械系统的模型设计。
- 对非线性关系进行线性化，以便产生线性模型。

18.1　工程系统

在第 17 章中，我们分别介绍了平动机械结构、转动机械结构、电气、流体和热系统的基本构建模型。然而，在工程中遇到的很多系统包含多种的学科领域。例如，一台电动机包含机械和电气两个方面的元件。本章关注于怎样把单个学科的内容组合起来，用以给出多学科系统的模型。同时，也尝试解决真实部件经常具有的非线性问题。例如，在考虑弹簧的简单模型时，无论有多么大的力，都假设力和形变成正比。该数学模型只是一个真实弹簧的简化。一般来说，非线性模型是更难以处理的，因此工程师都尝试回避它们，并且用一个线性模型来近似。

18.2　转动 - 平动系统

有很多机械涉及旋转运动到平移运动或平移运动到旋转运动的转换。例如，齿轮齿条、带丝杆的轴、线缆和滑轮系统等。

为了说明怎样分析这样的系统，考虑一个齿条和齿轮机构（见图 18-1）。齿轮的旋转运动可变换为齿条的平移运动。首先考虑齿轮。对其作用的净转矩为（$T_{in} - T_{out}$）。故考虑转动惯量，并假设忽略阻尼，有：

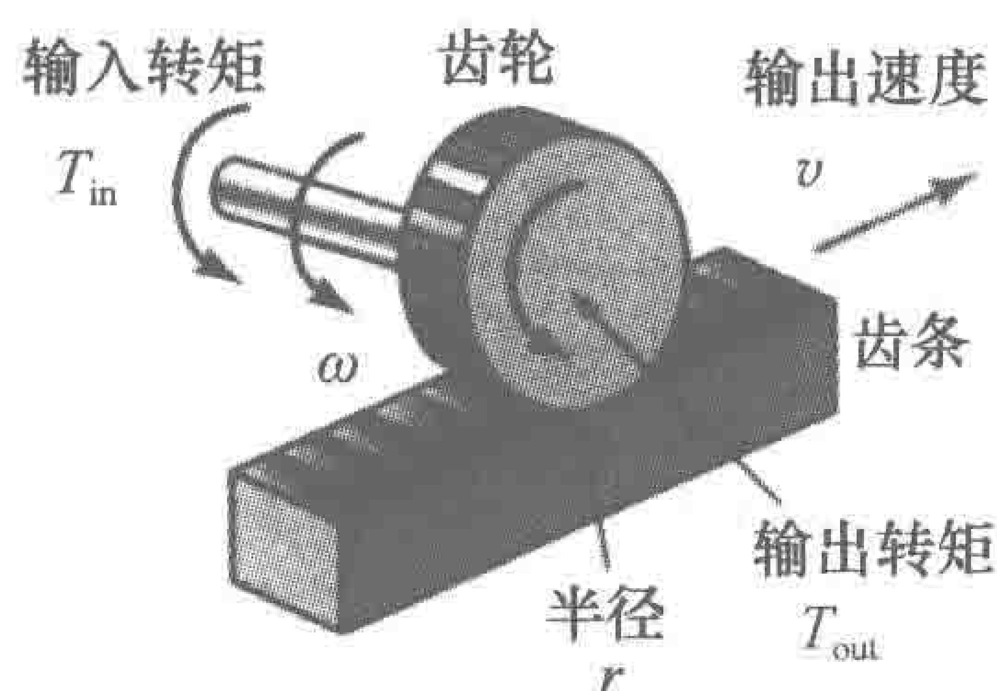


图 18-1　齿条和齿轮

$$T_{in} - T_{out} = I\frac{d\omega}{dt}$$

其中 I 是齿轮的转动惯量，ω 是角速度。齿轮的转动将导致齿条的平动速度 v。如果齿轮的半径为 r，则 $v = r\omega$。故我们可写为：

$$T_{in} - T_{out} = \frac{I}{r}\frac{dv}{dt}$$

现考虑齿条。由于齿轮的运动，有作用力 T/r 作用在齿条上。如果摩擦力为 cv，则净力为：

$$\frac{T_{out}}{r} - cv = m\frac{dv}{dt}$$

从这两个方程中消掉 T_{out}，有：

$$T_{in} - rcv = \left(\frac{I}{r} + mr\right)\frac{dv}{dt}$$

故：

$$\frac{dv}{dt} = \left(\frac{r}{I + mr^2}\right)(T_{in} - rcv)$$

该结果是一个描述输出与输入关系的一阶微分方程。

18.3　机电一体化系统

机电器件，例如电位器、电动机和发电机等，将电信号变换为旋转运动，反之亦然。本节讨论怎样能得到这样的系统模型。电位器的输入为转动，输出为电势差。电动机的输入为电势差，输出为轴的转动。发电机的输入为轴的转动，输出为电势差。

18.3.1　电位器

旋转电位器（见图 18-2）是一种分压器，故有

$$\frac{v_o}{V} = \frac{\theta}{\theta_{max}}$$

其中 V 是电位器的整个长度上的电势差，θ_{max} 是滑动触头从电位器的一端旋转到另一端扫过的整个角度。对应输入 θ 的输出是 v_o。

18.3.2　直流电动机

直流电动机用于把电气输入信号转换成机械输出信号，通过电动机电枢线圈的电流使电动机的轴旋转，进而带动负载旋转（见图 18-3）。

电动机主要由电枢线圈组成，它可自由旋转。电枢线圈处于由通过定子绕组的电流或永磁铁产生的磁场中。当电流 i_a 流过电枢线圈时，由于它处于磁场中，所以力作用于电枢线圈并使它旋转（见图 18-4）。处于与导线垂直的磁通密度为 B 的磁场中，且流过电流为 i_a 的长为 L 的导线的作用力 F 由 $F=Bi_aL$ 给出，对于 N 根导线力为 $F=NBi_aL$。作用到电枢线圈导线上的力产生转矩 T，即 $T=Fb$，其中 b 为线圈的宽度。故

$$T = NBi_aLb$$

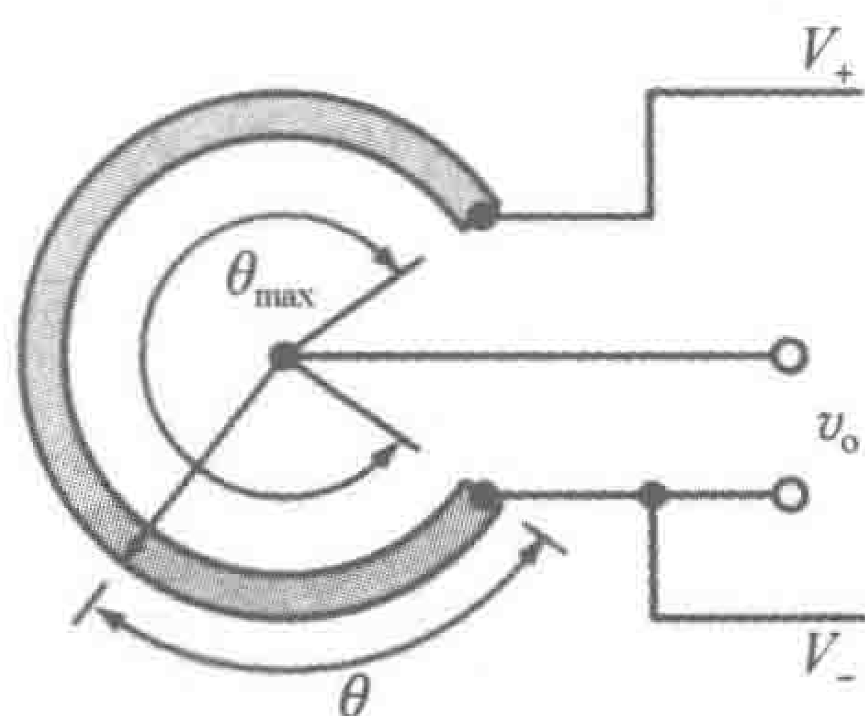


图 18-2　旋转电位器

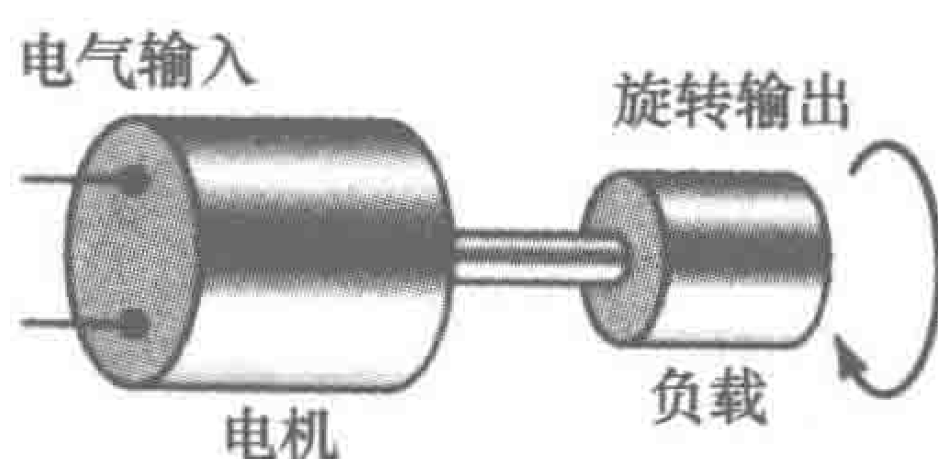


图 18-3　电动机驱动负载

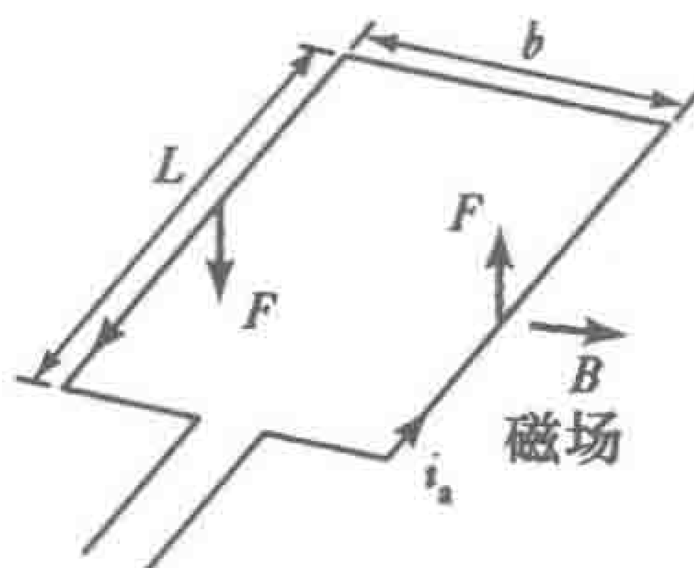


图 18-4　电枢线圈的一根导线

于是，最后得到的转矩与（Bi_a）成正比，其他的系数都是常量。因此我们可写为：

$$T = k_1Bi_a$$

由于电枢是在磁场中旋转的线圈，将有电压在线圈中因电磁感应而产生。该电压的方向与产生该电压的变化方向相反，称为反向电动势（e. m. f）。该反向电动势 v_b 与电枢的转速及线圈的磁通密度 B 成正比，故

$$v_b = k_2B\omega$$

其中 ω 为轴的角速度，k_2 是常量。

考虑一种直流电动机，它的电枢线圈和定子线圈分别激励。对于所谓的**电枢控制电动机**（armature-controlled motor），磁场电流 i_f 保持常量，电动机可通过调整电枢电压 v_a 来控制。固定的励磁电流意味着电枢线圈的固定的磁通密度 B。故

$$v_b = k_2B\omega = k_3\omega$$

其中 k_3 是常量。电枢电路可认为是电阻 R_a 和电感 L_a 的串联（见图 18-5）。

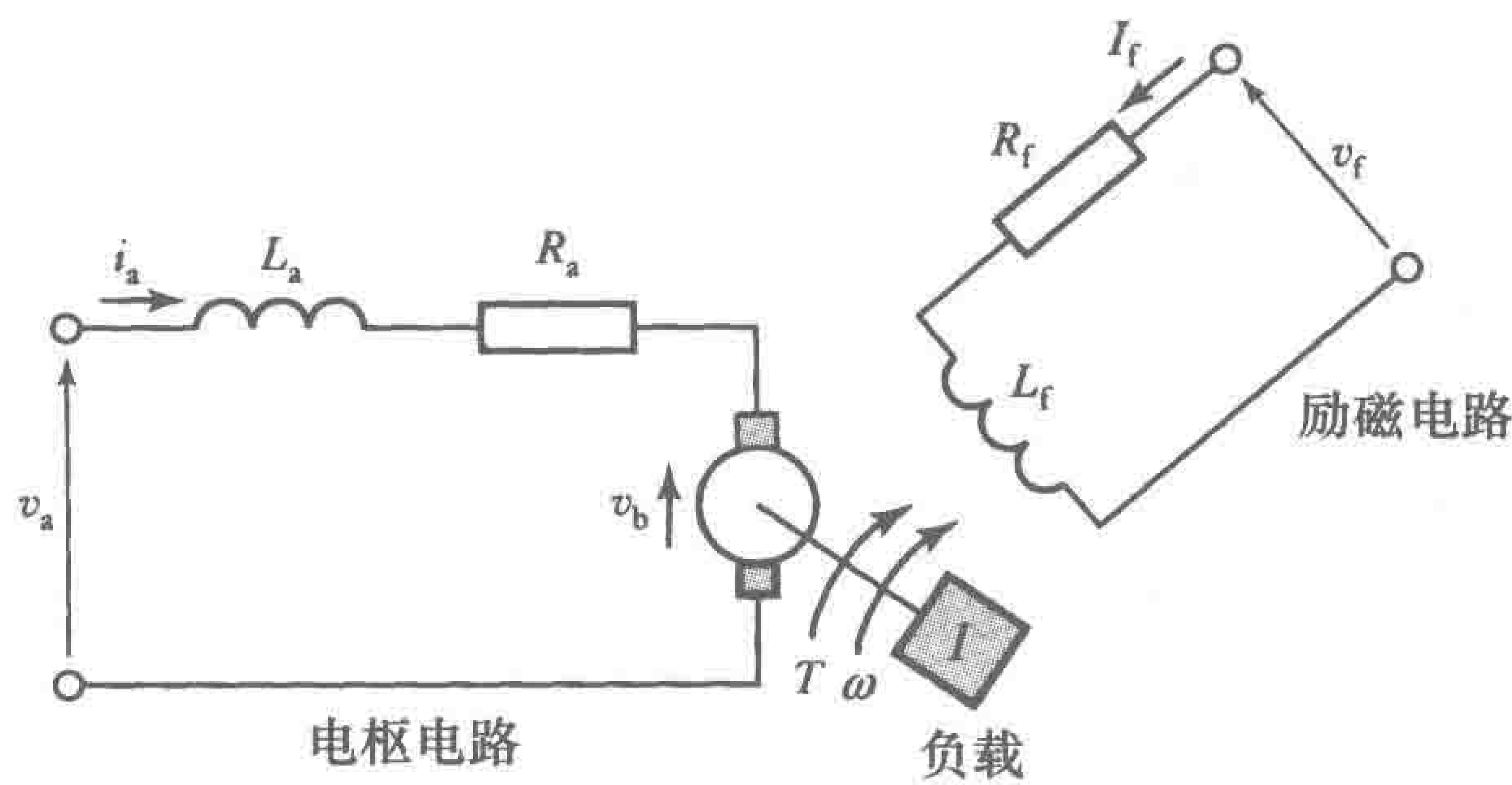


图 18-5 直流电动机电路

如果 v_a 是施加到电枢电路上的电压，由于存在反电动势 v_b，则我们有

440~441

$$v_a - v_b = L_a \frac{di_a}{dt} + R_a i_a$$

我们可通过图 18-6a 中的框图来考虑这一方程。系统的电动机部分的输入是 v_a，它与反电动势 v_b 的反馈信号的差给出一个误差信号，作为电枢电路的输入。因此，上述方程描述了对电枢线圈的误差信号的输入和电枢电流的输出 i_a 之间的关系。替换 v_b，有

$$v_a - k_3\omega = L_a \frac{di_a}{dt} + R_a i_a$$

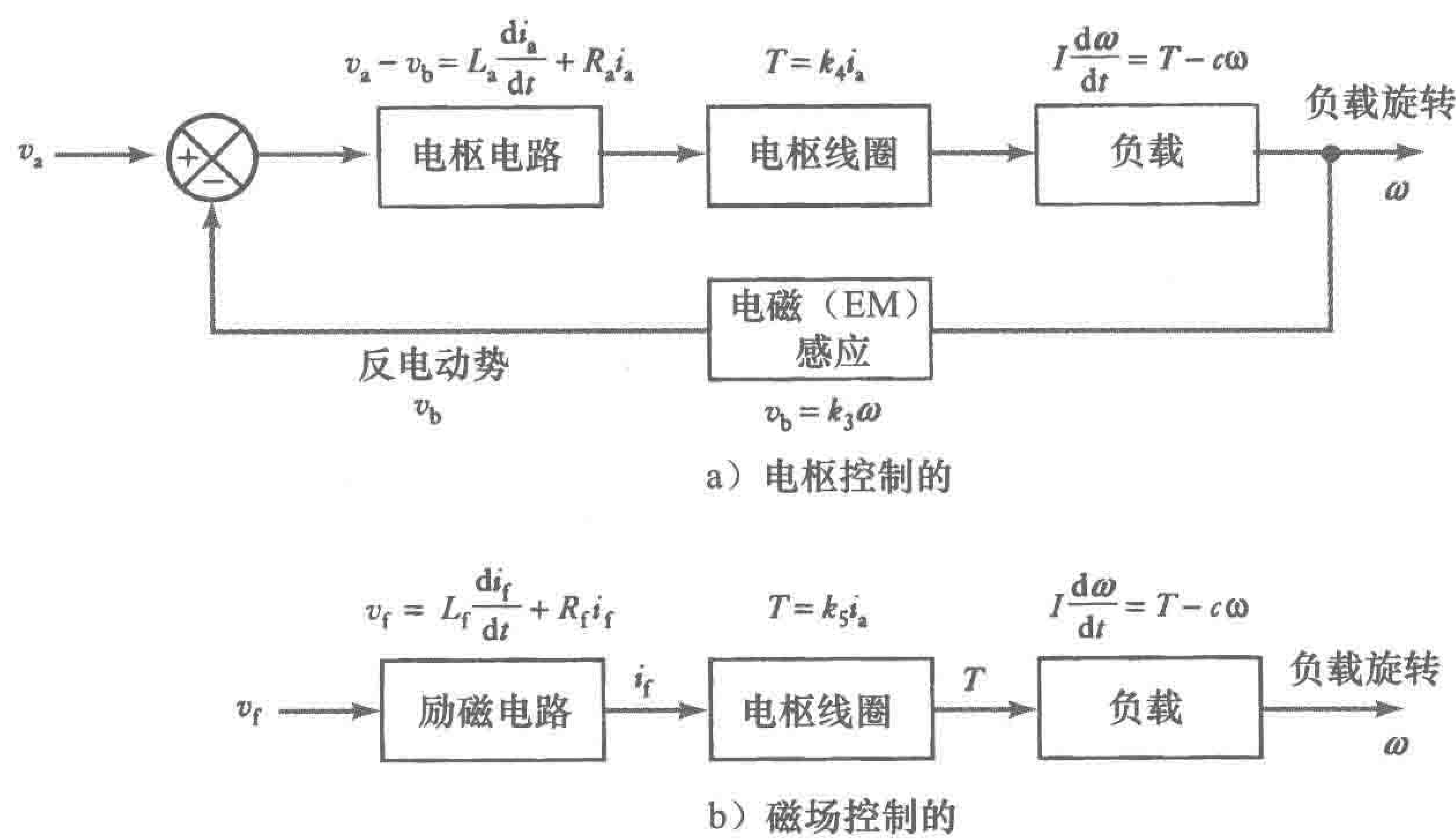


图 18-6 直流电动机

电枢中的电流 i_a 产生转矩 T。对于电枢控制的电动机，由于 B 是常量，所以我们有

$$T = k_1 B i_a = k_4 i_a$$

其中 k_4 是常量。于是，该转矩变为了负载系统的输入。作用于负载上的净转矩为：

$$净转矩 = T - 阻尼转矩$$

阻尼转矩为 $c\omega$，其中 c 是常量。因此，如忽略轴的扭转弹性的影响，则有

$$净转矩 = k_4 i_a - c\omega$$

该转矩将引起角加速度$\frac{d\omega}{dt}$，故

$$I\frac{d\omega}{dt} = k_4 i_a - c\omega$$

因此，我们可用两个方程来描述电枢控制的电动机符合的条件，即

442

$$v_a - k_3\omega = L_a\frac{di_a}{dt} + R_a i_a \quad 和 \quad I\frac{d\omega}{dt} = k_4 i_a - c\omega$$

因此我们可通过消掉 i_a 获得输出 ω 与系统输入 v_a 相关的方程。请见第 20 章或附录 A 中拉普拉斯变换的简短讨论，了解这一变换如何实现的详细信息。

对于所谓的**磁场控制电动机**，电枢电流保持常量，而通过改变磁场电压来控制电动机。对于励磁电路（见图 18-5），基本上只是电感 L_f 与电阻 R_f 串联。故，对于该电路有：

$$v_f = R_f i_f + L_f\frac{di_f}{dt}$$

我们可根据图 18-6b 中所示的框图来考虑磁场控制的电动机。系统的输入是 v_f。励磁电路把该电压转换成电流 i_f，v_f 与 i_f 之间的关系见上式。该电流使磁场产生，因此有一个转矩作用到电枢线圈，由 $T=k_1Bi_a$ 给出。但是磁通密度 B 与励磁电流 i_f 成正比，而 i_a 是常量，故：

$$T = k_1Bi_a = k_5 i_f$$

其中，k_5 是常量。然后该转矩输出被负载系统转换成角速度 ω。如前一样，作用到负载上的净转矩将为：

$$净转矩 = T - 阻尼转矩$$

阻尼转矩为 $c\omega$，其中 c 是常量。因此，如果忽略轴的扭转弹性的影响，则有

$$净转矩 = k_5 i_f - c\omega$$

该转矩将引起角加速度$\frac{d\omega}{dt}$，故

$$I\frac{d\omega}{dt} = k_5 i_f - c\omega$$

因此，磁场控制的电动机符合的条件被方程描述为：

$$v_f = R_f i_f + L_f\frac{di_f}{dt} \quad 和 \quad I\frac{d\omega}{dt} = k_5 i_f - c\omega$$

故我们可通过消掉 i_f 获得输出 ω 与系统输入 v_f 相关的方程。请见第 20 章或附录 A 中拉普拉斯变换的简短讨论，了解这一变换如何实现的详细信息。

18.4　线性度

在结合框图来建立系统模型的过程中，我们假设每个框的关系是线性的。由于很多实际的工程项目是非线性的，所以下面简要讨论线性度以及我们需要怎样对一个非线性项目进行线性近似。

对于一个理想弹簧来说，力 F 和产生的伸展长度 x 之间的关系是线性的，由 $F=kx$ 给出。这意味着如果力 F_1 产生伸展长度 x_1，力 F_2 产生伸展长度 x_2，则力（F_1+F_2）产生伸展长度（x_1+x_2）。该原理称为**叠加原理**，是**线性系统**的一个必要条件。线性系统的另一个条件是，如果一个输入力 F_1 产生伸展长度 x_1，则输入 cF_1 将产生输出 cx_1，其中 c 是倍数。

443

当力 F 与伸展长度 x 的关系是线性时，其曲线图是经过原点的直线（见图 18-7a）。实际的弹簧，像任何其他的实际部件一样，不是完全线性的（见图 18-7b）。然而，往往在一定的操作范围内可以假定是线性的。因此，对于图 18-7b 中的弹簧的曲线图，只有在弹簧曲线的中间部分可以假定是线性的。对于很多系统组件，在某些操作点附近的变量值的范围内的操作，可以假设是线性的。

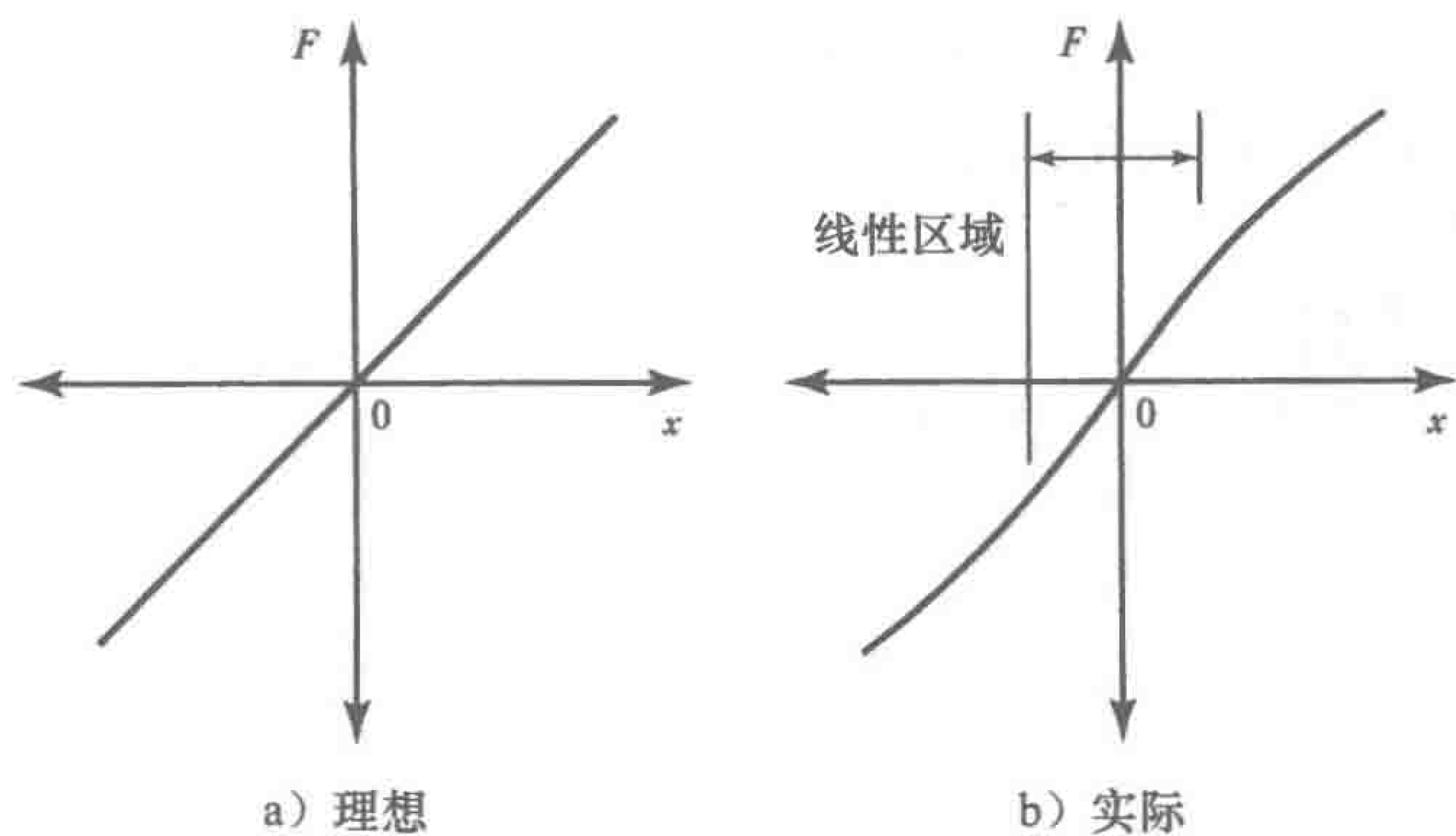


图 18-7　弹簧

对于某些系统部件（见图 18-8a）输入/输出关系是非线性的。对于这样的部件，取得线性关系的最好方法就是使部件在工作点处曲线的切线所在的直线上工作。

因此对于图 18-8a 中 y 和 x 之间的关系，在工作点 P 处的斜率值为 m

$$\Delta y = m\Delta x$$

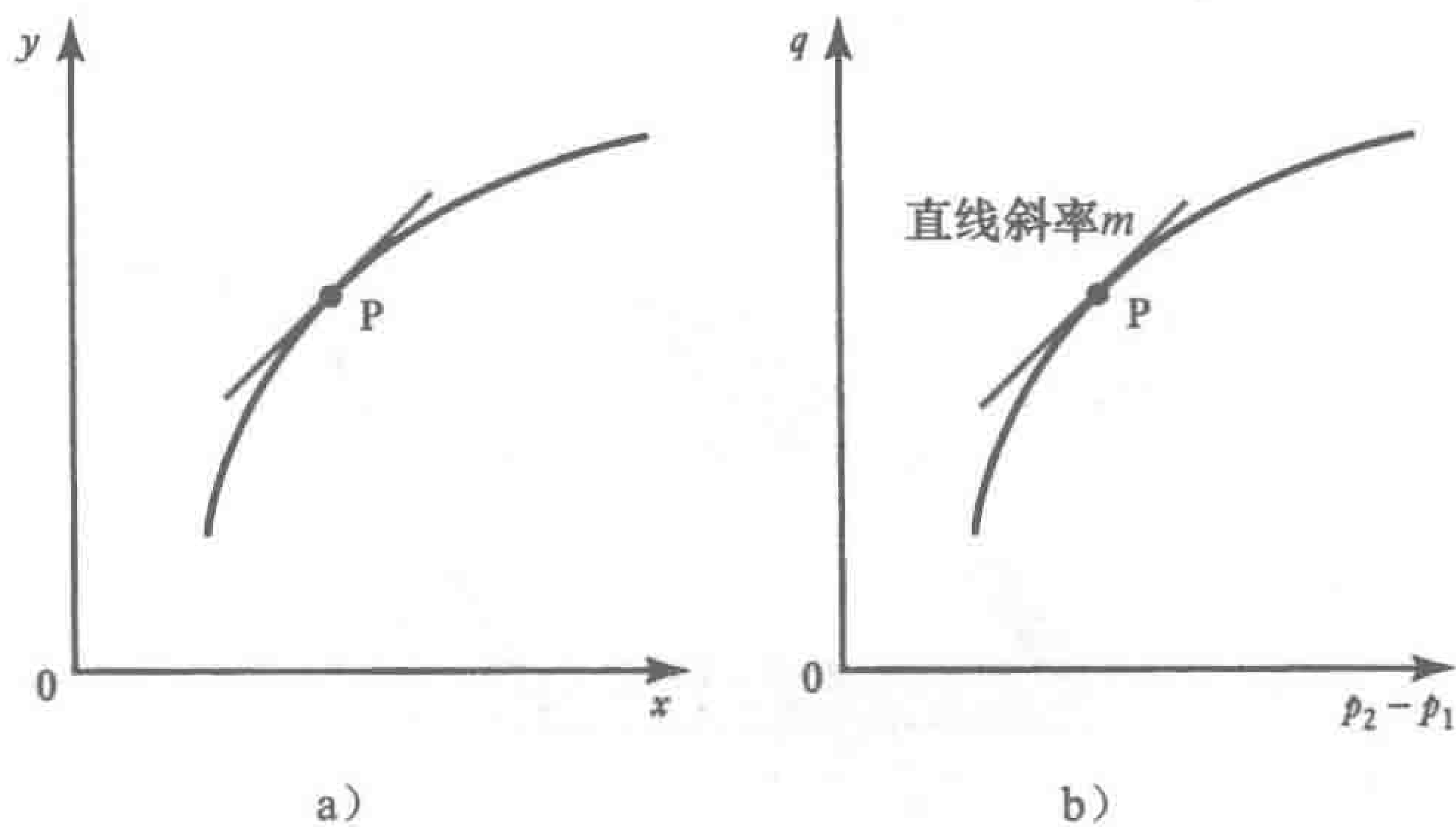


图 18-8　非线性关系

其中 Δy 和 Δx 是工作点附近输入和输出信号的微小变化量。

例如，通过孔的液体的流量 q 由下式给出：

$$q = c_d A \sqrt{\frac{2(p_1 - p_2)}{\rho}}$$

444

其中 c_d 是一个称为流量系数的常量，A 是孔的横截面积，ρ 是液体密度，$(p_1 - p_2)$ 是压力差。对于固定的横截面积和液体密度，方程可写为：

$$q = C\sqrt{p_1 - p_2}$$

其中 C 是常量。这是流量和压力差之间的非线性关系。我们可以通过考虑在工作点的代表流量、压力差曲线的斜率的直线，获得一个线性关系（见图 18-8b）。斜率 m 是 $dq/d(p_1 - p_2)$，且值为：

$$m = \frac{dq}{d(p_1 - p_2)} = \frac{C}{2\sqrt{p_{o1} - p_{o2}}}$$

其中 $(p_{o1} - p_{o2})$ 是工作点上的值。对于工作点附近很小的改变，我们假设可用斜率为 m 的直线来代替非线性曲线，因此可写 m 为 $m = \Delta q/\Delta(p_1 - p_2)$，故：

$$\Delta q = m\Delta(p_1 - p_2)$$

因此，如果每 kPa 有 $C = 2\text{m}^3/\text{s}$，即 $q = 2\sqrt{p_1 - p_2}$，则对于 $(p_1 - p_2) = 4\text{kPa}$ 的工作点，$m = 2/(2\sqrt{4}) = 0.5$，方程的线性化描述为

$$\Delta q = 0.5\Delta(p_1 - p_2)$$

由于大多数控制系统技术是基于这些系统的元器件存在线性关系的假设，所以使用线性化的数学模型。而且，由于大多数控制系统保持输出为某个参考值，与该值相比变化非常小，所以该线性化模型是相当近似。

18.5　液压机械系统

液压机械的转换装置包括液压信号到平动或旋转运动的变换，或相反。例如，由液体压力使液压缸内的活塞运动，这个过程就包括系统的液压输入到平动运动输出的变换。

图 18-9 说明一个液压系统。在系统中，一个位移输入 x_i 在经过系统之后，被转换成一个负载的位移 x_0。该系统由滑阀和液压缸组成。向左边的位移输入 x_i 使流体在供给压力 p_s 的作用下流入到液压缸的左侧。推动液压缸内的活塞向右运动，使腔体右侧的流体通过滑阀的右端出口流出。

液体进入和流出腔体的流量取决于打开端口使流体进入或离开滑阀的输入运动。当输入位445移 x_i 向右时，滑阀使液体流到液压缸的右端，导致液压缸内的活塞向左运动。

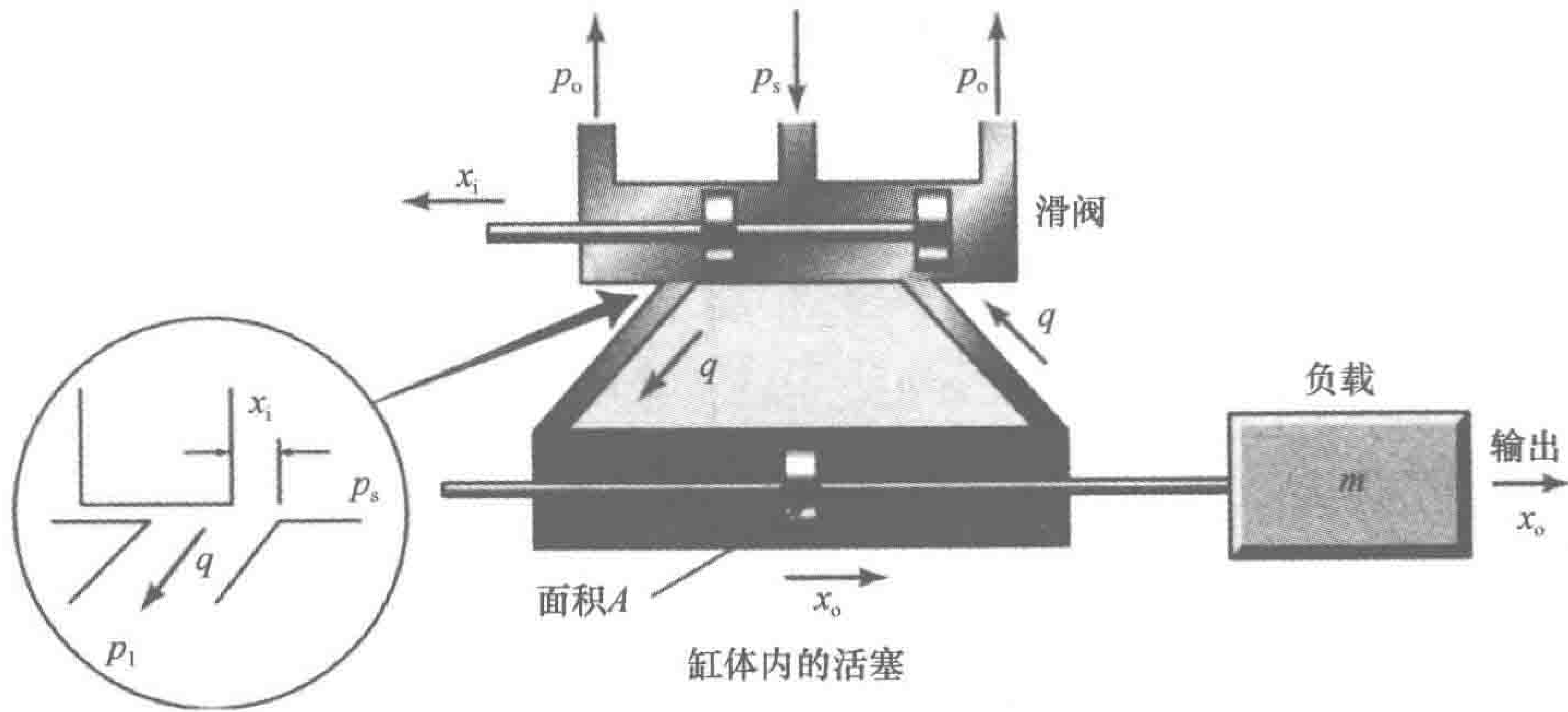


图 18-9　液压系统和负载

液体通过滑阀端口（即小孔）的流量 q 取决于孔两侧之间的压力差和它的横截面积 A 之间的非线性关系。然而，可以使用一个方程的线性化形式（见前一节的推导）：

$$\Delta q = m_1\Delta A + m_2\Delta(\text{压力差})$$

其中 m_1 和 m_2 是工作点上的常量。对于进入腔体内的流体，压力差为 $(p_s - p_1)$；而对于出口，压力差为 $(p_2 - p_o)$。如果在工作点附近方程被线性化，且把该点作为滑阀的中央点，则连接滑阀与缸体的端口就关闭了，那么这种条件下的 q 是 0，故 $\Delta q = 0$。如果从中央位置测量的距离是 x_i，则 A 与 x_i 成正比。而且在活塞的入口侧相对于 p_s 的压力改变为 $-\Delta p_1$，而在出口侧相对于 p_o 的压力改变为 Δp_2。因此，对于输入端口方程可写为：

$$q = m_1 x_i + m_2(-\Delta p_1)$$

对于输出端口为：

$$q = m_1 x_i + m_2\Delta p_2$$

两个方程相加，有

$$2q = 2m_1 x_i - m_2(\Delta p_1 - \Delta p_2)$$

$$q = m_1 x_i - m_3(\Delta p_1 - \Delta p_2)$$

其中 $m_3 = m_2/2$。

对于液压缸，当活塞移动的距离是 x_o 时，进入腔体左边或右边的液体的体积变化是 Ax_o，其中 A 是活塞的横截面积。因此，体积的变化率为 $A(\mathrm{d}x_o/\mathrm{d}t)$。进入液压缸左侧的液体流量是 q。然而，由于从活塞的一端到另一端存在流体的泄漏流量，因此有

446

$$q = A\frac{\mathrm{d}x_o}{\mathrm{d}t} + q_L$$

其中 q_L 是泄漏流量。替换 q 给出：

$$m_1 x_i - m_3(\Delta p_1 - \Delta p_2) = A\frac{\mathrm{d}x_o}{\mathrm{d}t} + q_L$$

泄漏流量 q_L 是指通过活塞和液压缸之间的缝隙孔的流量。等截面是常量，压力差为（$\Delta p_1 - \Delta p_2$）。因此，对于该流体应用线性化方程：

$$q_L = m_4(\Delta p_1 - \Delta p_2)$$

故，用这一方程替换 q_L，有

$$m_1 x_i - m_3(\Delta p_1 - \Delta p_2) = A\frac{\mathrm{d}x_o}{\mathrm{d}t} + m_4(\Delta p_1 - \Delta p_2)$$

$$m_1 x_i - (m_3 + m_4)(\Delta p_1 - \Delta p_2) = A\frac{\mathrm{d}x_o}{\mathrm{d}t}$$

活塞两边的压力差导致一个施加到负载上的力，该力为（$\Delta p_1 - \Delta p_2$）A。然而，存在质量的运动阻尼，即摩擦，它是与质量的速度 $\mathrm{d}x_o/\mathrm{d}t$ 成正比的。因此，作用到负载上的净力为：

$$\text{净力} = (\Delta p_1 - \Delta p_2)A - c\frac{\mathrm{d}x_o}{\mathrm{d}t}$$

该净力引起质量加速，该加速度为（$\mathrm{d}^2x_o/\mathrm{d}t^2$），故

$$m\frac{\mathrm{d}^2x_o}{\mathrm{d}t^2} = (\Delta p_1 - \Delta p_2)A - c\frac{\mathrm{d}x_o}{\mathrm{d}t}$$

重新整理给出：

$$\Delta p_1 - \Delta p_2 = \frac{m}{A}\frac{\mathrm{d}^2x_o}{\mathrm{d}t^2} + \frac{c}{A}\frac{\mathrm{d}x_o}{\mathrm{d}t}$$

用该方程替换原来方程中的压力差，

$$m_1 x_i - (m_3 + m_4)\left(\frac{m}{A}\frac{\mathrm{d}^2x_o}{\mathrm{d}t^2} + \frac{c}{A}\frac{\mathrm{d}x_o}{\mathrm{d}t}\right) = A\frac{\mathrm{d}x_o}{\mathrm{d}t}$$

重新整理给出：

$$\frac{(m_3 + m_4)m}{A}\frac{\mathrm{d}^2x_o}{\mathrm{d}t^2} + \left(A + \frac{c(m_3 + m_4)}{A}\right)\frac{\mathrm{d}x_o}{\mathrm{d}t} = m_1 x_i$$

447

重新整理该方程给出：

$$\frac{(m_3 + m_4)m}{A^2 + c(m_3 + m_4)}\frac{\mathrm{d}^2x_o}{\mathrm{d}t^2} + \frac{\mathrm{d}x_o}{\mathrm{d}t} = \frac{Am_1}{A^2 + c(m_3 + m_4)}x_i$$

该方程可通过引入两个常量 k 和 τ 被简化，后一个常量称为时间常量（见第 19 章）。因此，有

$$\tau\frac{\mathrm{d}^2x_o}{\mathrm{d}t^2} + \frac{\mathrm{d}x_o}{\mathrm{d}t} = kx_i$$

因此输入和输出之间的关系可用一个二阶微分方程描述。

本章小结

我们在工程中遇到的很多系统，其组成需要多个学科的内容。我们可通过分别分析单个学科的功能模块并进行组合的方法来建立整个系统。

无论系统的基本方程是代数方程还是微分方程，其输出的幅度直接是由输入按比例产生的，则称为线性的。对于代数方程，这意味着输出对输入的曲线是通过原点的直线。因此，如果输入加倍，则输出也加倍。对于线性系统，我们可将分别考虑的每一单独输入产生的系统输出进行累加，获得系统对一系列输入的输出，这称为**叠加原理**。

习题

18.1 请推导关于直流伺服电动机的输入电压和输出角速度的微分方程，假设电动机是电枢控制的，且电动机的等效电路只有电阻，电感被忽略。

18.2 请推导一个直流发电机的微分方程。可假设该发电机有一恒定磁场。该电枢电路有电枢线圈，包括电阻和电感，与负载串联。假设负载有电阻和电感。

448

18.3 请推导永磁直流电动机的微分方程。

第 19 章　系统的动态响应

本章目的

通过本章的学习，读者将能够：

- 通过微分方程对动态系统进行建模。
- 确定一阶系统的输出对输入的响应和时间常量。
- 确定二阶系统的输出对输入的响应，并识别欠阻尼、临界阻尼和过阻尼的条件。
- 根据上升时间、超调量、衰减率、减量和建立时间描述二阶系统响应的特性。

19.1　动态系统建模

测量或控制系统模型设计的重要功能是能针对特定的输入将产生什么输出进行预测。我们不只是关心经过一段时间，对应于输入 y 的输出 x 已达到稳态的静态情况。我们必须考虑当输入发生变化或者输入是时变信号时，输出将怎样随时间变化。例如，当温控器被设置到一个新的温度时，温控系统的温度将怎样随着时间变化？对于控制系统，当设定为一个新输入值或可能增加稳定速度时，系统的输出将怎样随着时间变化？

第 17 章和第 18 章分别讨论输入为时变信号时的系统模型和用微分方程来表示系统响应。本章是关于如何应用模型来对时变输入对应的时变输出进行预测。

19.1.1　微分方程

为了描述系统输入和输出之间的关系，我们必须描述输入和输出之间关于时间的函数关系。因此，我们需要公式来表达：当系统输入随着时间变化时，输出怎样随时间变化。这可以通过微分方程来实现。这些方程包含对时间的导数，因此给出了系统响应随时间变化的信息。导数 dx/dt 描述 x 随时间变化的速度；导数 d^2x/dt^2 描述 dx/dt 如何随时间变化。根据方程中导数的最高阶数，微分方程可分为一阶、二阶、三阶等。对于一阶方程来说，最高阶数是 dx/dt；对于二阶、三阶来说是 d^2x/dt^2 和 d^3x/dt^3；对于 n 阶方程来说，最高阶数是 d^nx/dt^n。

本章所讲述的内容主要是关于一阶、二阶系统的响应类型和由系统不同类型的输入所得响应对应差分方程的解。本章应用“试解法”来求解方程，而拉普拉斯变换法将在第 20 章进行介绍。

19.2　术语

本节我们将关注用于描述系统动态响应的术语。

19.2.1　自然响应与受迫响应

自然响应是指当系统没有使系统变量改变的输入，而只是发生自然变化时的系统响应。如图 19-1a所示，考虑一阶系统中水自然从箱体流出的情况。

对于这样一个系统，有：

$$p_1 - p_2 = Rq$$

R 是液阻。同时，$p_1 - p_2 = h\rho g$，其中 ρ 是水的密度，q 是水流出水箱的速度，即 $-dV/dt$，V 是水箱内水的体积，记为 Ah。因此，$q = -d(Ah)/dt = -Adh/dt$，上述方程可写为：

$$h\rho g = -RA\frac{dh}{dt}$$

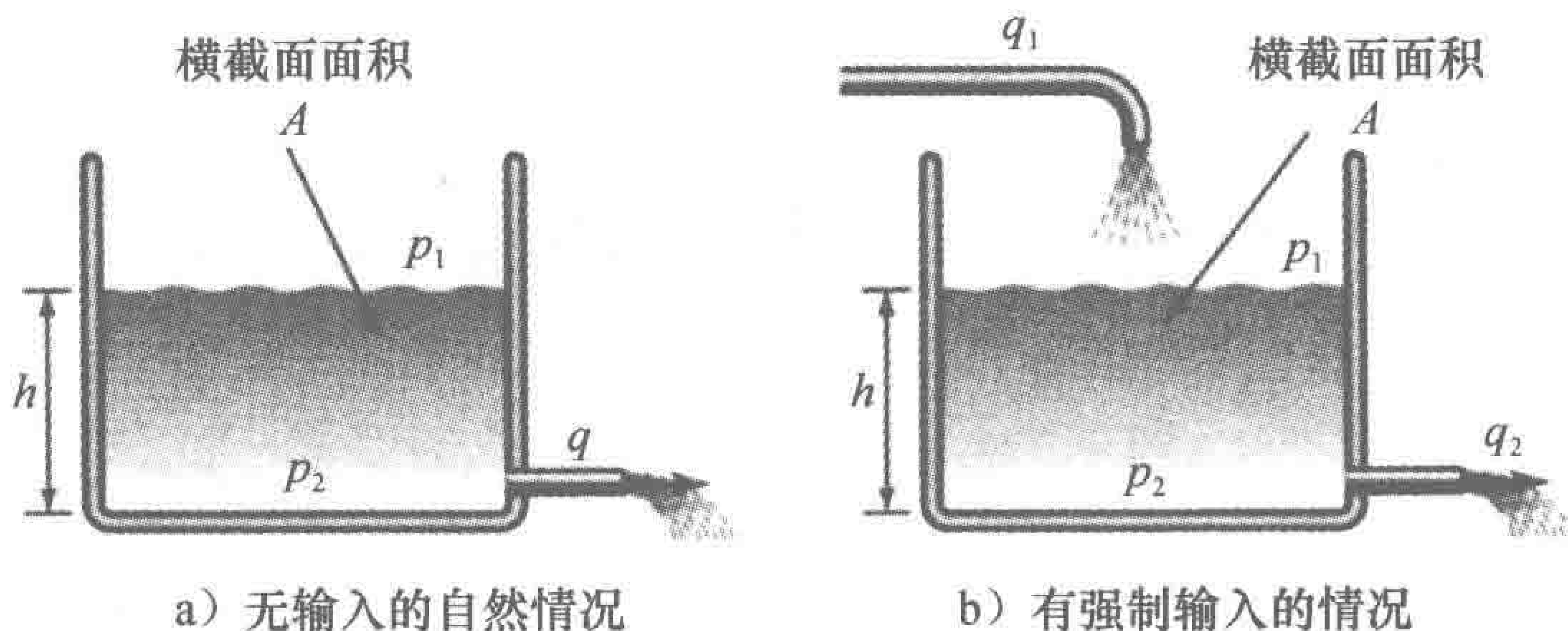


图 19-1 水从箱体流出

该式对应无任何输入时系统变量 h 的自然响应，它只是随着时间自然改变。我们可以注意该微分方程，其中所有 h 的输出项在等号的左侧，零输入项在右侧，即

450

$$RA\frac{dh}{dt} + (\rho g)h = 0$$

17.4.1 节中推导出了水箱的微分方程，不仅有水从水箱中流出，而且还有水流入水箱（见图 19-1b）。这个方程中有一个强制输入函数 q_1，可以写成下面的形式：

$$RA\frac{dh}{dt} + (\rho g)h = q_1$$

作为另外一个例子，考虑放入温度为 T_L 的热的液体中温度计。17.5.1 节中推导出了温度计的读数 T 以一定的速率随时间变化的情况，并由如下的微分方程给出：

$$RC\frac{dT}{dt} + T = T_L$$

这个微分方程有个强制输入 T_L。

19.2.2 瞬态与稳态响应

控制系统的总响应或系统中某个部分的响应是由稳态响应与瞬态响应两个部分组成的。**瞬态响应**是指由于输入改变而产生的系统响应部分，经过很短的时间间隔响应结束。而**稳态响应**是在所有的瞬态响应结束后系统所保持的响应。

为了简单地说明这一点，考虑一个垂直悬挂弹簧（见图 19-2），看看当重物突然悬挂到弹簧上的时候会发生什么。弹簧的变形突然增加，然后可能产生振荡，直到一段时间后达到一稳定值。该稳定值是弹簧系统的稳态响应。在稳态之前发生的震荡是瞬态响应。

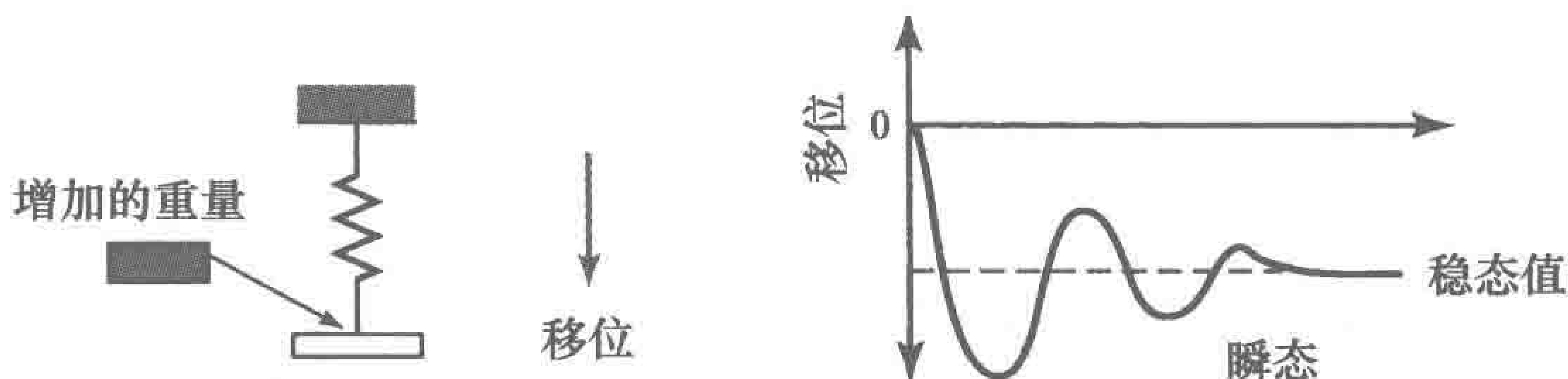


图 19-2 弹簧系统的瞬态与稳态响应

19.2.3 输入的形式

上述弹簧系统的输入，即重量，是一个随时间变化的量。在某些特定的时间弹簧没加上任何重量，即没有输入，之后的时间将保持一个恒定的输入。这种输入类型称为**阶跃输入**（step input），如图 19-3a 所示。

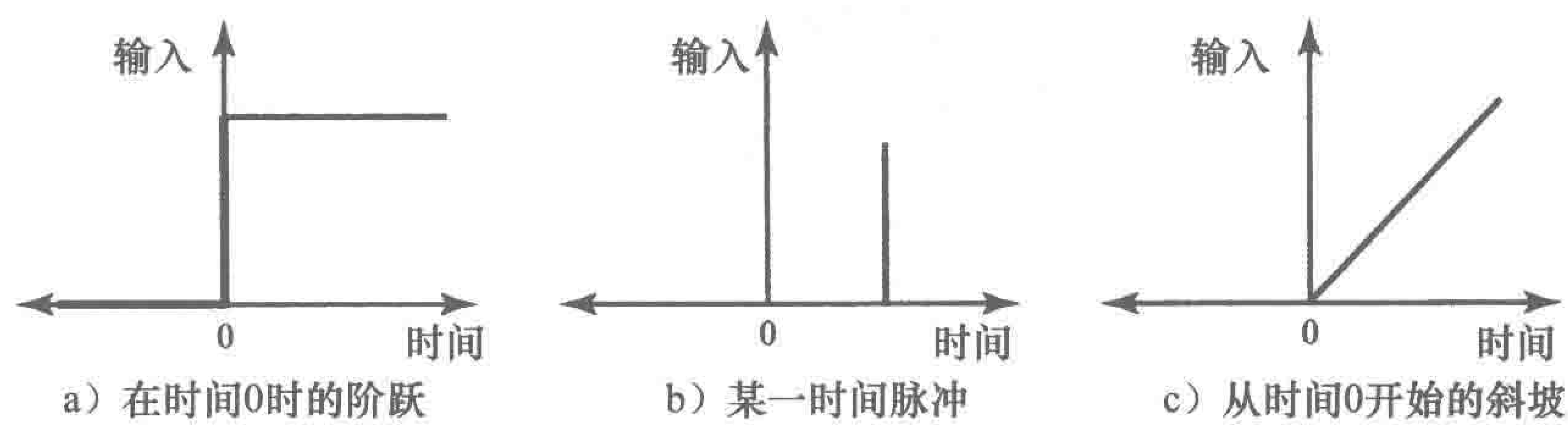


图 19-3　输入的形式

系统的输入信号还可采用其他的形式，例如，脉冲、斜坡和正弦信号。**脉冲**信号是非常短时间的输入（见图 19-3b）；**斜坡**信号是一个逐渐增加的输入（见图 19-3c），可表示为 $y = kt$，其中 k 是常量；而**正弦**信号输入可表示为 $y = k\sin\omega t$ 描述，其中 ω 称为角频率，等于 $2\pi f$，f 是频率。

输入/输出都是时间的函数。因此它们可写为 $f(t)$ 的形式，其中 f 是函数，(t) 是指函数值依赖于时间 t。于是，对于弹簧系统的重量输入 W，我们可以写为 $W(t)$，而对于变形 d 输出可写为 $d(t)$。$y(t)$ 是通常用于输入，$x(t)$ 通常用于输出。

19.3　一阶系统

考虑一阶系统，输入为 $y(t)$，输出为 $x(t)$，强制输入为 b_0y，于是该系统可由如下微分方程来描述：

$$a_1\frac{\mathrm{d}x}{\mathrm{d}t} + a_0x = b_0y$$

其中 a_1、a_0、b_0 是常量。

19.3.1　自然响应

输入 $y(t)$ 有很多形式。考虑第一种情况，即输入为零。由于系统没有输入，所以除了自然响应外，没有任何信号受迫系统做出响应，那么系统的微分方程为：

$$a_1\frac{dx}{\mathrm{d}t} + a_0x = 0$$

我们可以通过**变量分离**的方法来解这个方程。把所有与 x 变量相关的部分写在等号左边，与 t 变量相关的写在右边。

$$\frac{\mathrm{d}x}{x} = -\frac{a_0}{a_1}\mathrm{d}t$$

针对单位的阶跃输入，从 $t=0$ 时 $x=1$ 的初始值，到 t 时 x 值之间进行积分，得到：

$$\ln x = -\frac{a_0}{a_1}t$$

于是：

$$x = \mathrm{e}^{-a_0t/a_1}$$

451
~
452

然而，我们已认识到，可能有一个微分方程，其解的形式为 $x = A\mathrm{e}^{st}$，其中 A 和 s 是常量。于是有 $\mathrm{d}x/\mathrm{d}t = sA\mathrm{e}^{st}$，替代微分方程中的这些值，得到：

$$a_1sA\mathrm{e}^{st} + a_0A\mathrm{e}^{st} = 0$$

$a_1s + a_0 = 0$，$s = -a_0/a_1$。于是，解是：

$$x = A\mathrm{e}^{-a_0t/a_1}$$

因为没有受迫响应，所以这个解称为自然响应。我们可以通过给出某些初始的边界条件来确定常量 A 的值。因此，如果 $t=0$ 时 $x=1$，则 $A=1$。图 19-4 给出了自然响应，即指数衰减：

$$x = e^{-a_0 t/a_1}$$

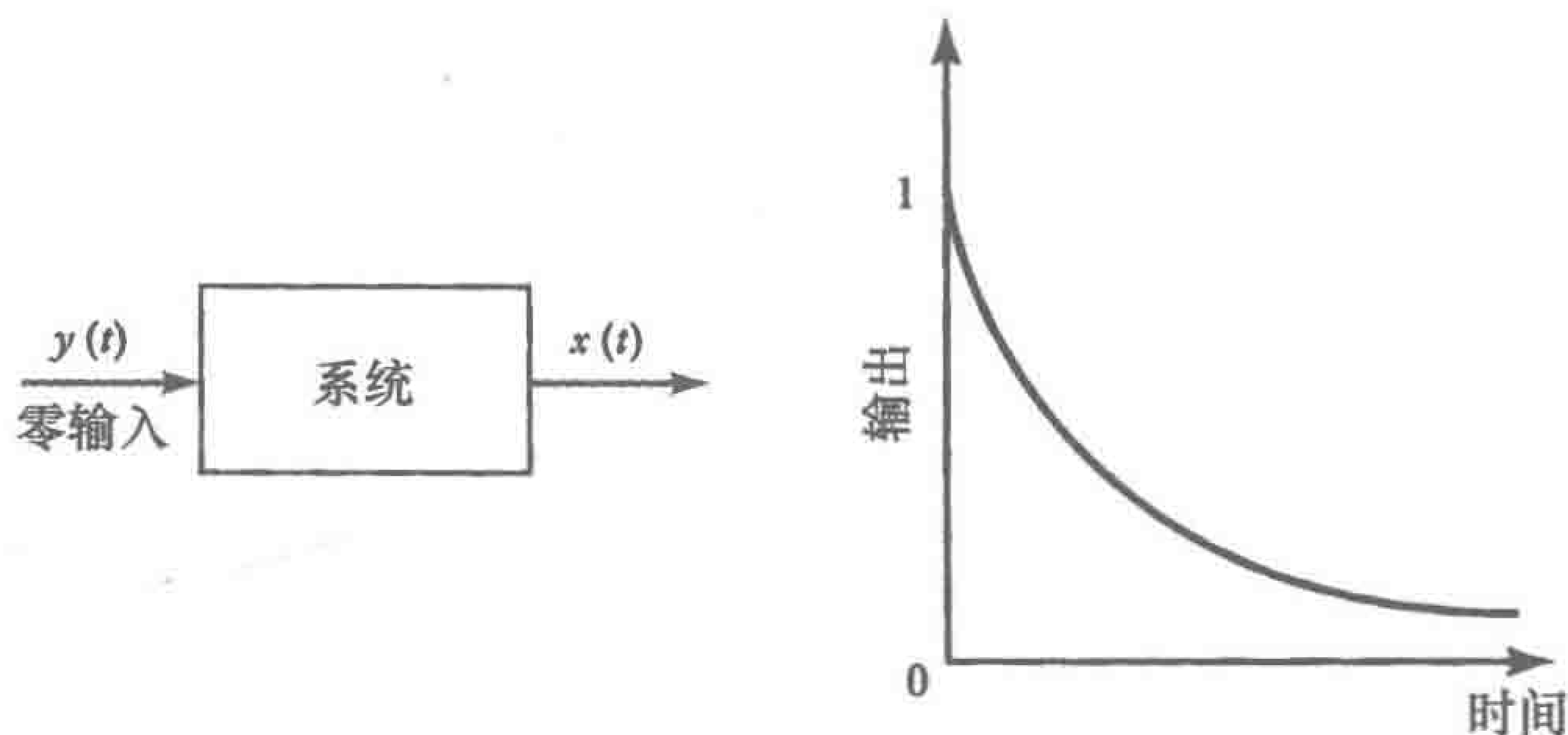


图 19-4 一阶系统的自然响应

19.3.2 受迫输入响应

现考虑如下有**受迫函数**的微分方程：

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

这个方程的解是由两部分组成的，即 $x=u+v$。一部分代表解的瞬态部分，另一部分代表稳态部分。把该式带入微分方程，可得：

$$a_1 \frac{d(u+v)}{dt} + a_0(u+v) = b_0 y$$

重新整理该式可得：

$$\left(a_1 \frac{du}{dt} + a_0 u\right) + \left(a_1 \frac{dv}{dt} + a_0 v\right) = b_0 y$$

如果令

$$a_1 \frac{dv}{dt} + a_0 v = b_0 y$$

则有

$$a_1 \frac{du}{dt} + a_0 \mu = 0$$

453

于是，两个微分方程中，一个包含受迫响应，而另一个只是自然响应。后一个方程只是本征方程，我们已经在本节的前面给出了它的解，于是我们可以得到如下形式的解：

$$u = Ae^{-a_0 t/a_1}$$

另一个微分方程包含受迫函数 y。对于这个微分方程解的形式，我们尝试依赖于输入信号 y 的形式。对于阶跃输入来说，y 是常量，且一直大于 0，即 $y=k$，我们可以尝试一个解 $v=A$，其中 A 是常量。如果我们有 $y=a+bt+ct^2+\cdots$形式的输入信号，其中 a，b，c 是可为 0 的常量，那么我们可以尝试 $v=A+Bt+Ct^2+\cdots$形式的解。对于正弦信号，我们可尝试 $v=A\cos\omega t+B\sin\omega t$ 形式的解。

为了说明这一点，假设在 $t=0$ 时有一个大小为 k 的阶跃输入（见图 19-5a），我们尝试一个 $v=A$ 形式的解。对常量进行微分的结果是 0，那么当这个形式的解被代入微分方程中时，将有 $a_0 A=b_0 k$，于是 $v=(b_0/a_0)\ k$。

完整解将由 $x=u+v$ 给出，于是有

$$x = A\mathrm{e}^{-a_0t/a_1} + \frac{b_0}{a_0}k$$

通过给出初始的（边界）条件我们可确定常量 A 的值。于是，当 $t=0$ 时，输出 $x=0$，有

$$0 = A + \frac{b_0}{a_0}k$$

于是，$A = -(b_0/a_0)k$。那么方程的解变为：

$$x = \frac{b_0}{a_0}k(1 - \mathrm{e}^{-a_0t/a_1})$$

当 $t\to\infty$ 时，指数项趋向于0。指数项给出了响应的瞬态解部分。稳态响应是当 $t\to\infty$ 时的 x 值，即 $(b_0/a_0)k$。于是方程可写为：

$$x = 稳态值 \times (1 - \mathrm{e}^{-a_0t/a_1})$$

图 19-5b 给出了阶跃响应随时间变化的曲线图。

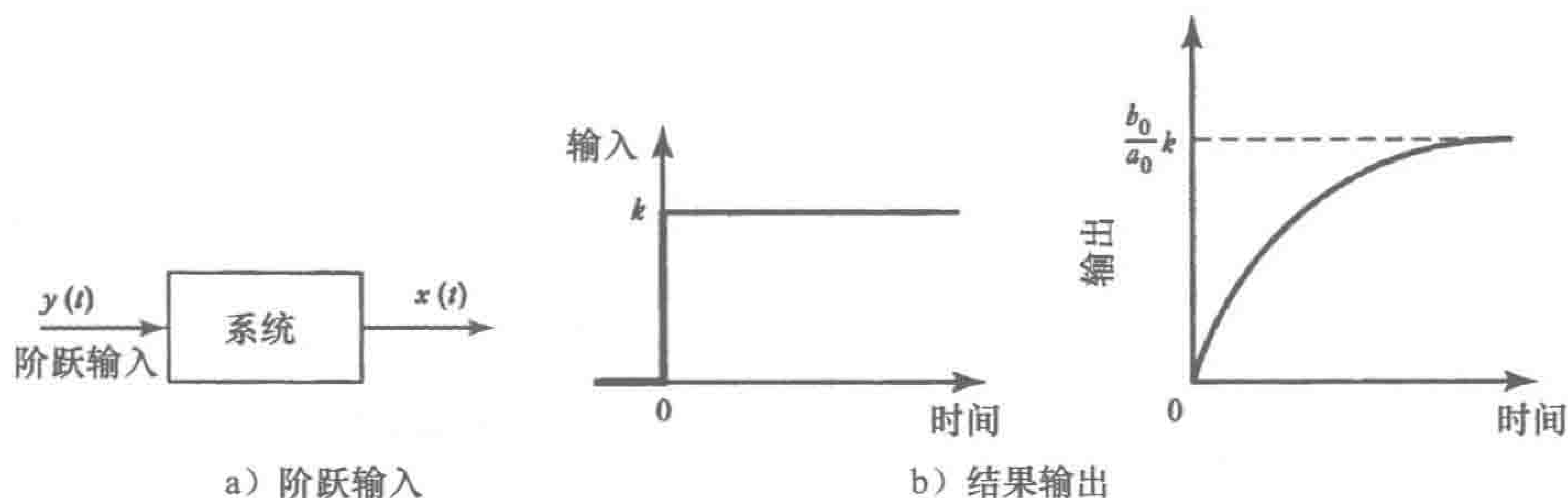


图 19-5　阶跃输入与响应

19.3.3　一阶系统的例子

作为以上论述的进一步说明，考虑下面一阶系统的例子。

一个电气转换系统由一个电阻和一个电容串联组成。当有阶跃输入 V 时，电容两端得到电势差为 v 的输出。v 值由如下微分方程给出：

$$RC\frac{\mathrm{d}v}{\mathrm{d}t} + v = V$$

将该微分方程与前面解出来的方程比较，有：$a_1 = RC$，$a_0 = 1$，$b_0 = 1$。于是，解的形式为：

$$v = V(1 - \mathrm{e}^{-t/RC})$$

现在考虑一个 1MΩ 电阻和 1 个 2μF 的电容组成的电路。在时刻 $t=0$ 电路遭受一个 $4t$V 的斜坡电压，即电压以每秒 4V 的速度增加。该微分方程与之前给出的例子有类似的形式，但由斜坡电压 $4t$ 替换了之前例子的阶跃电压，即

$$RC\frac{\mathrm{d}v}{\mathrm{d}t} + v = 4t$$

于是，带入上面给出的值得到，

$$2\frac{\mathrm{d}v}{\mathrm{d}t} + v = 4t$$

令 $v = v_n + v_f$，即本征和受迫响应之和。对于自然响应，我们有

$$2\frac{\mathrm{d}v_n}{\mathrm{d}t} + v_n = 0$$

对于受迫响应，有

$$2\frac{dv_f}{dt}+v_f=4t$$

对于自然响应微分方程，我们可用 $v_n=Ae^{st}$形式的解进行尝试，有

$$2Ase^{st}+Ae^{st}=0$$

于是，$s=-1/2$，$v_n=Ae^{-t/2}$。

对于受迫响应微分方程，由于方程的右边 $4t$，我们可尝试 $v_f=A+Bt$ 形式的解。代入这个式子有 $2B+A+Bt=4t$。于是可得 $B=4$，$A=-2B=-8$。因此，解为 $v_f=-8+4t$。于是整体解为：

$$v=v_n+v_f=Ae^{-t/2}-8+4t$$

454～455

由于当 $t=0$ 时 $v=0$，所以有 $A=8$。于是：

$$v=8e^{-t/2}-8+4t$$

作为进一步的例子，考虑一台电动机，电动机的输出角速度 ω 和输入电压 v 之间的关系由下式给出：

$$\frac{IR}{k_1k_2}\frac{d\omega}{dt}+\omega=\frac{1}{k_1}v$$

比较该微分方程与上面求解的方程，有 $a_1=IR/k_1k_2$，$a_0=1$，$b_0=1/k_1$。1V 的阶跃输入的稳态值是 $(b_0/a_0)=1/k_1$。

19.3.4　时间常量

当一个一阶系统得到大小为 k 的阶跃输入时，我们可以根据下式得到随时间 t 变化的输出：

$$x=\frac{b_0}{a_0}k(1-e^{-a_0t/a_1})$$

或

$$x=\text{稳态值}\times(1-e^{-a_0t/a_1})$$

当时间 $t=(a_1/a_0)$ 时，上式的指数项为 $e^{-1}=0.37$，且

$$x=\text{稳态值}\times(1-0.37)$$

在这段时间里，输出上升到了稳态值的 0.63 倍。这一时间称为**时间常量** τ：

$$\tau=\frac{a_1}{a_0}$$

在 $2(a_1/a_0)=2\tau$ 的时间里，指数项变为 $e^{-2}=0.14$，因此：

$$x=\text{稳态值}\times(1-0.14)$$

这时输出上升到稳态值的 0.86 倍。用类似的方法可以计算出经过 3τ、4τ、5τ 等时间后的输出值。表 19-1 给出了计算结果，图 19-6 给出了单位阶跃响应随时间变化的曲线。

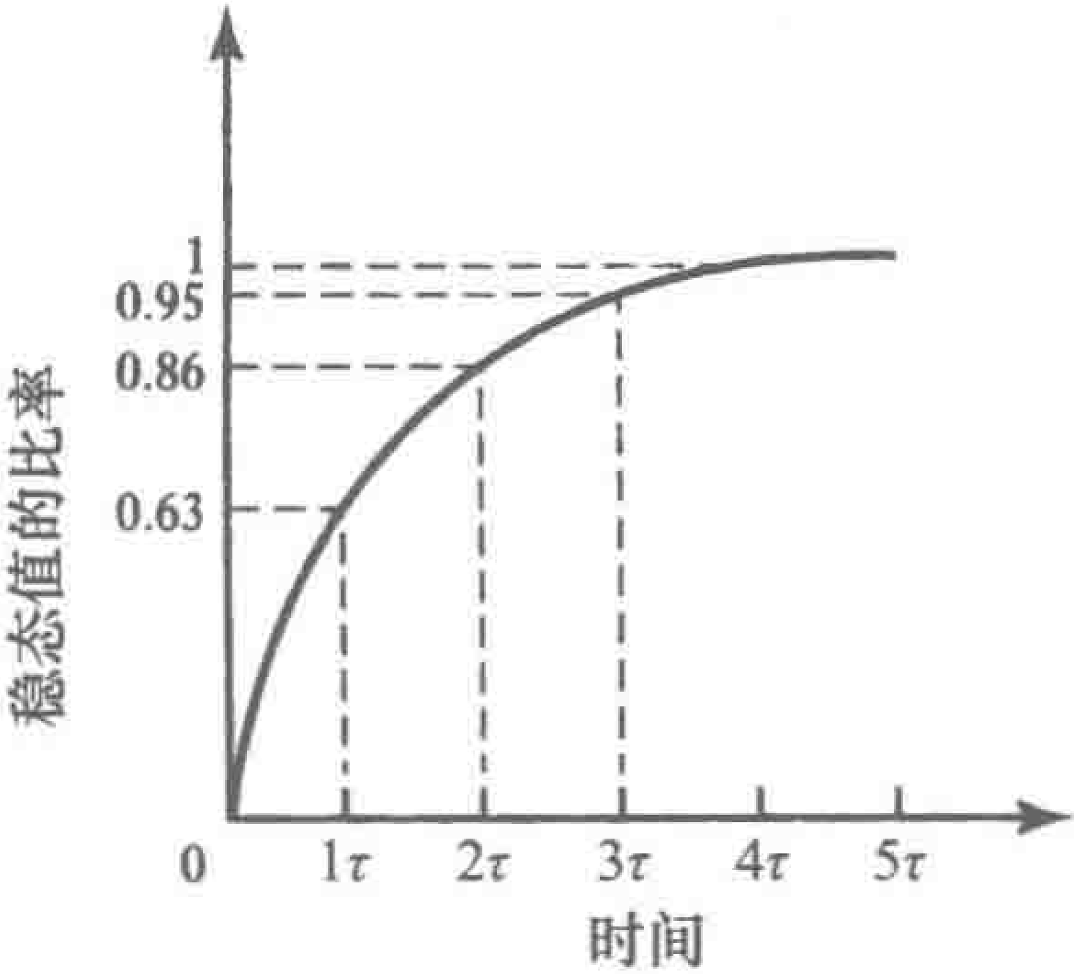


图 19-6　一阶系统阶跃响应

表 19-1　一阶系统阶跃响应

时间 t	稳态输出的比率	时间 t	稳态输出的比率
0	0	4τ	0.98
1τ	0.63	5τ	0.99
2τ	0.86	∞	1
3τ	0.95		

根据时间常量 τ，我们可以写出描述一阶系统响应的方程：

$$x = 稳态值 \times (1 - e^{-t/\tau})$$

时间常量 τ 是 a_1/a_0，于是我们可以写出一阶微分方程的通式为：

$$a_1 \frac{dx}{dt} + a_0 x = b_0 y$$

或

$$\tau \frac{dx}{dt} + x = \frac{b_0}{a_0} y$$

但 b_0/a_0 是与输入 y 相乘得到稳态值的系数。我们称之为**稳态增益**（steady-state gain），因为该因子说明在稳态条件下输出比输入大多少。这样，如果我们用 G_{SS} 表示该因子，微分方程可写为如下形式：

$$\tau \frac{dx}{dt} + x = G_{SS} y$$

为了说明这一点，请考虑图 19-7。图中给出了当一阶系统输入 5V 的阶跃信号时，输出 v_o 随时间变化的情况。时间常量是指一阶系统的输出从 0 变到最终稳态值的 0.63 倍所用的时间。在这个例子中，时间常量大约是 3s。这是一个一阶系统，我们可以通过寻找 2 倍的时间常量值，即 6s，来检查这个值。对于一阶系统来说，这时的输出应该是稳态值的 0.86 倍。在本例中，它正是这样。稳态输出是 10V。于是，稳态增益 G_{SS}（稳态输出/输入）= 10/5 = 2。一阶系统的微分方程可写成：

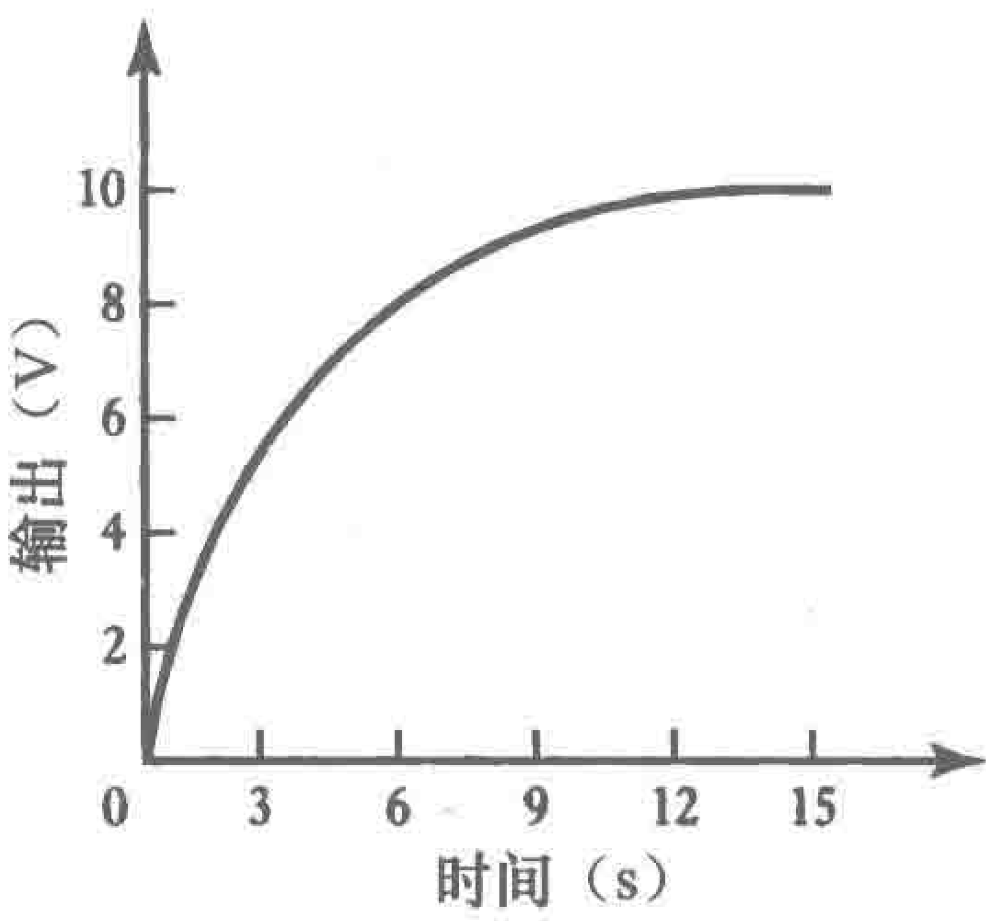


图 19-7　例子

$$\tau \frac{dx}{dt} + x = G_{SS} y$$

于是，对于该系统，我们有：

$$3 \frac{dv_o}{dt} + v_o = 2v_i$$

19.4　二阶系统

很多二阶系统本质上可当作类似于一个带有质量和阻尼的绷紧弹簧。图 19-8 给出了这一系统的基本原理。

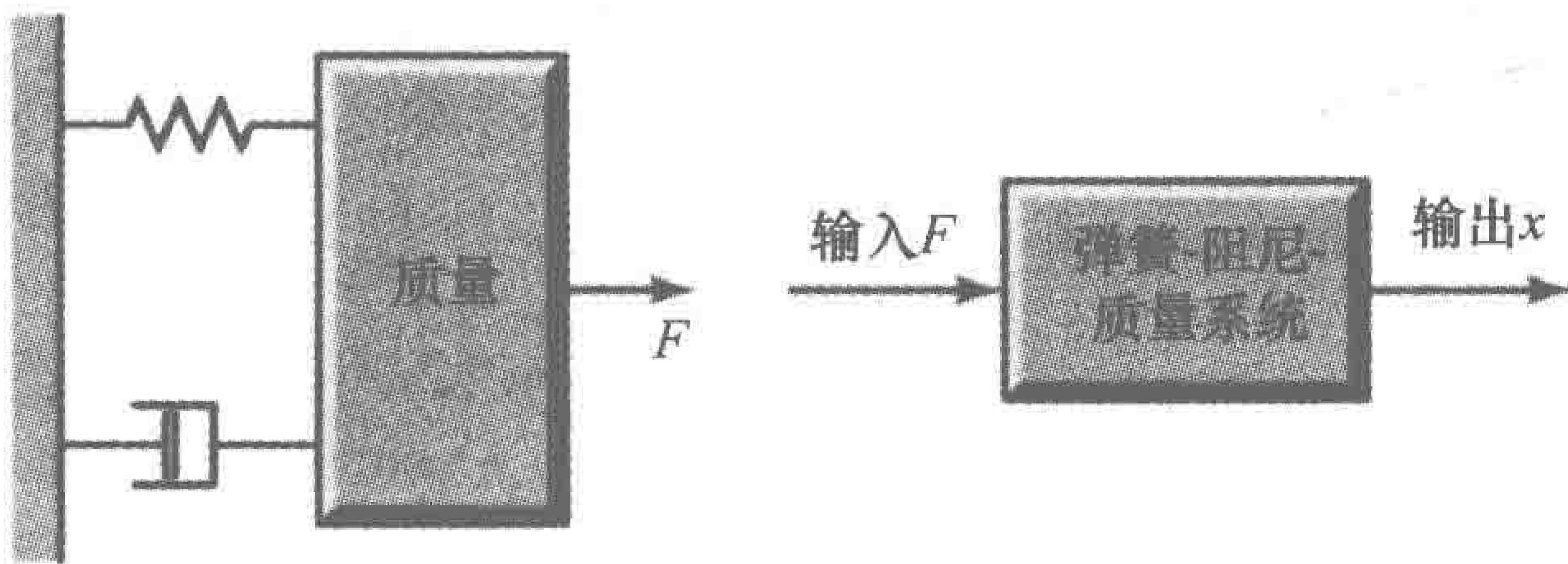


图 19-8　弹簧 - 阻尼 - 质量系统

这样的系统已在 17.2.2 节中进行了分析。描述输入力 F 和输出位移 x 之间关系的方程是：

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F$$

其中 m 是质量，c 是阻尼常量，k 是弹簧常量。

产生的位移 x 以何种方式随时间变化，将取决于系统中阻尼的大小。这样，如果施加的力

是阶跃输入，且没有任何阻尼，则质量块将自由地在弹簧上振荡并一直持续下去。无阻尼意味着 $c=0$，因此 $\mathrm{d}x/\mathrm{d}t$ 项为0。然而，阻尼会导致振荡消失，直到获得质量块的稳定位移。如果阻尼足够高，则不会有振荡产生，该质量块的位移将随着时间慢慢增加，逐渐地移动到它的稳定位置。图 19-9 给出了在不同大小的阻尼下，阶跃输入产生的位移随着时间变化的一般情况。

456 ~ 458

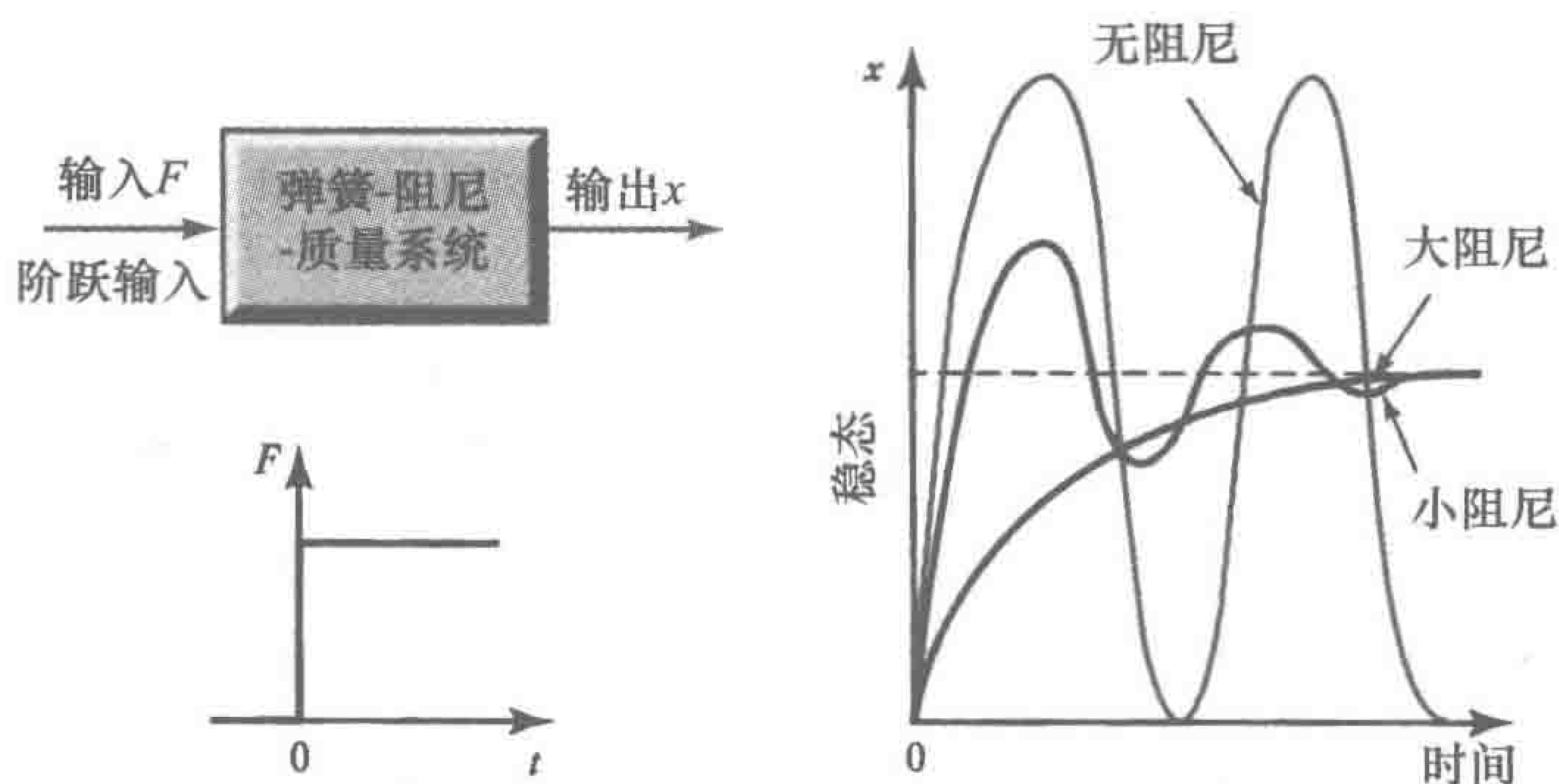


图 19-9　二阶系统的阻尼效果

19.4.1　自然响应

考虑在弹簧末端的一个质量块。

在没有任何阻尼，且没有受迫输入的情况下，二阶系统的输出是连续振荡（简谐运动）。这样，假设我们用如下方程描述这个振荡：

$$x = A\sin\omega_{\mathrm{n}}t$$

其中，x 是在时刻 t 时的位移，A 是振荡的振幅，ω_{n} 是自由无阻尼振荡的角频率。对时间求一阶微分有：

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \omega_{\mathrm{n}}A\cos\omega_{\mathrm{n}}t$$

二阶微分有：

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = -\omega_{\mathrm{n}}^2A\sin\omega_{\mathrm{n}}t = -\omega_{\mathrm{n}}^2x$$

整理上式，可以得到微分方程：

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \omega_{\mathrm{n}}^2x = 0$$

但对于刚度为 k 的弹簧上的质量块 m，恢复力为 kx，于是：

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} = -kx$$

上式可写成：

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{k}{m}x = 0$$

这样，比较这两个微分方程，有：

$$\omega_{\mathrm{n}}^2 = \frac{k}{m}$$

459

$x=A\sin\omega_{\mathrm{n}}t$ 是该微分方程的解。

现在考虑有阻尼的情况。质量块的运动可描述为：

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + c\frac{\mathrm{d}x}{\mathrm{d}t} + kx = 0$$

为了解这个方程，我们可尝试形式为 $x_n = Ae^{st}$ 的解。于是有 $dx_n/dt = Ase^{st}$ 和 $d^2x_n/dt^2 = As^2e^{st}$。这样，把这些值代入微分方程得出：

$$mAs^2e^{st} + cAse^{st} + kAe^{st} = 0$$

$$ms^2 + cs + k = 0$$

于是，$x_n = Ae^{st}$ 可以作为上述方程等于 0 的解。这一方程称为**辅助方程**。方程的根可通过因式分解或二次方程的求根公式来得到，于是：

$$s = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \frac{k}{m}}$$

$$= -\frac{c}{2m} \pm \sqrt{\frac{k}{m}\left(\frac{c^2}{4mk}\right) - \frac{k}{m}}$$

但是 $\omega_n{}^2 = k/m$，所以如果令 $\zeta^2 = c^2/4mk$，那么上述方程可以写成：

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

其中 ζ 称为**阻尼因子**。

从上式得到的 s 值依赖于平方根项。这样，当 ζ^2 大于 1 时，平方根项给出了正数的平方根，当 ζ^2 小于 1 时，得到的是负数平方根。阻尼因子决定了这个平方根项是正数还是负数，因此系统的输出形式如下给出。

1. 过阻尼

当 $\zeta > 1$ 时，有两个不同的实数根 s_1 和 s_2：

$$s_1 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$$

$$s_2 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$

于是 x_n 的一般解为：

$$x_n = Ae^{s_1t} + Be^{s_2t}$$

系统在这种情况下称为**过阻尼**（over-damped）。

2. 临界阻尼

当 $\zeta = 1$ 时，有两个相等的实数根 $s_1 = s_2 = -\omega_n$。在这种情况下，系统称为**临界阻尼**（critically damped）。

$$x_n = (At + B)e^{-\omega_nt}$$

从上式可以看出，这种情况下解的形式应该类似于 $x_n = Ae^{st}$，而且这个解的形式需要两个常量。 [460]

3. 欠阻尼

当 $\zeta < 1$ 时，由于这些根都包含（−1）的平方根，所以有两个复数根：

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} = -\zeta\omega_n \pm \omega_n\sqrt{-1}\sqrt{1 - \zeta^2}$$

把（−1）的平方根写为 j，

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

如果我们令

$$\omega = \omega_n\sqrt{1 - \zeta^2}$$

那么可以把 s 写为 $-\zeta\omega_n \pm j\omega$，因此两个根为：

$$s_1 = -\zeta\omega_n + j\omega \text{ 和 } s_2 = -\zeta\omega_n - j\omega$$

当系统处在由 ζ 决定的阻尼条件下时，ω 是运动的角频率。这些条件下的解是：

$$x_n = Ae^{(-\zeta\omega_n + j\omega)t} + Be^{(-\zeta\omega_n - j\omega)t} = e^{-\zeta\omega_nt}(Ae^{j\omega t} + Be^{-j\omega t})$$

但 $e^{j\omega t} = \cos\omega t + j\sin\omega t$ 和 $e^{-j\omega t} = \cos\omega t - j\sin\omega t$。因此：

$$x_n = e^{-\zeta\omega_nt}(A\cos\omega t + jA\sin\omega t + B\cos\omega t - jB\sin\omega t)$$

$$= e^{-\zeta\omega_n t}[(A+B)\cos\omega t + j(A-B)\sin\omega t)]$$

如果我们用常量 P 和Q代替（$A+B$）和 $j(A-B)$，则

$$x_n = e^{-\zeta\omega_n t}(P\cos\omega t + Q\sin\omega t)$$

在这样的条件下系统称为**欠阻尼**（under-damped）。

19.4.2 有受迫输入时的响应

当有一个受迫输入 F 时，微分方程变为：

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = F$$

我们可以使用前面解一阶微分方程的方法来解这个二阶微分方程，且方程的解是由瞬态（本征）响应和受迫响应两部分组成，即 $x=x_n+x_f$。替换上式中的 x，则有：

$$m\frac{d^2(x_n+x_f)}{dt^2} + c\frac{d(x_n+x_f)}{dt} + k(x_n+x_f) = F$$

如果令：

461

$$m\frac{d^2x_n}{dt^2} + c\frac{dx_n}{dt} + kx_n = 0$$

则有：

$$m\frac{d^2x_f}{dt^2} + c\frac{dx_f}{dt} + kx_f = F$$

前一节给出了方程解中自然响应部分的解。为了解这个受迫方程，

$$m\frac{d^2x_f}{dt^2} + c\frac{dx_f}{dt} + kx_f = F$$

我们需要考虑输入信号的一种特殊形式，然后尝试一个解。这样，当时间 $t=0$ 时阶跃输入为 F，我们可尝试 $x_f=A$ 这个解，其中 A 是常量（见 19.3.2 节中关于选择一阶微分方程解的讨论）。那么 $dx_f/dt=0$ 和 $d^2x_f/dt^2=0$。因此，当把这些值代入微分方程时，有 $0+0+kA=F$，所以 $A=F/k$，$x_f=F/k$。于是，对于过阻尼系统的本征和受迫响应之和组成的完整解为：

$$x = Ae^{s_1t} + Be^{s_2t} + \frac{F}{k}$$

对于临界阻尼系统的解为：

$$x = (At+B)e^{-\omega_n t} + \frac{F}{k}$$

而对于欠阻尼系统的解为：

$$x = e^{-\zeta\omega_n t}(P\cos\omega t + Q\sin\omega t) + \frac{F}{k}$$

当 $t\to\infty$ 时，上述三个方程的解都将趋向于 $x=F/k$。这就是**稳态条件**（steady-state condition）。于是，二阶微分方程

$$a_2\frac{d^2x}{dt^2} + a_1\frac{dx}{dt} + a_0x = b_0y$$

的自然频率为：

$$\omega_n^2 = \frac{a_0}{a_2}$$

而阻尼因子为：

$$\zeta^2 = \frac{a_1^2}{4a_2a_0}$$

19.4.3　二阶系统的例子

下面的例子说明了上面提出的知识点。

考虑一个 $R=100\Omega$，$L=2.0\mathrm{H}$，$C=20\mu\mathrm{F}$ 的串联 RLC 电路（见图 19-10）。当有一个阶跃输入 V 时，电路中电容 C 的两端的电势差 v_C 由下式给出（见与图 17-9 相关的文字）：

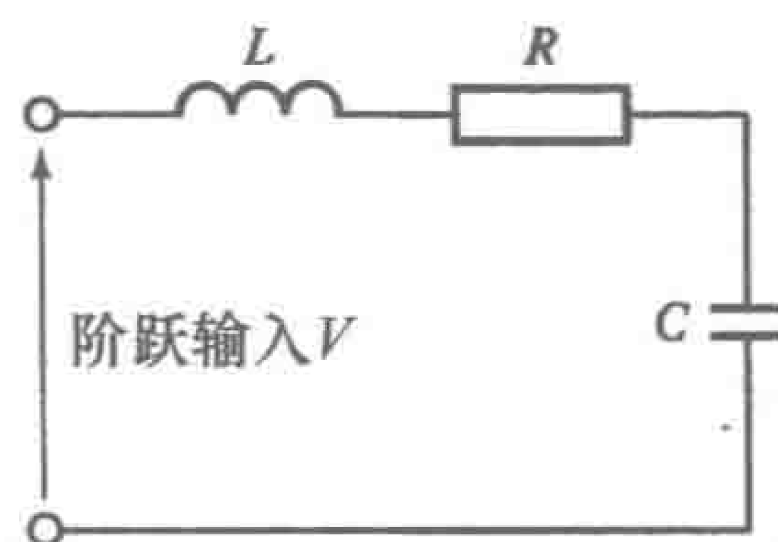


图 19-10　RLC 系统

$$\frac{d^2v_C}{dt^2}+\frac{R}{L}\frac{dv_C}{dt}+\frac{1}{LC}v_C=\frac{V}{LC}$$

如果我们把该方程与一般的二阶微分方程

$$a_2\frac{d^2x}{dt^2}+a_1\frac{dx}{dt}+a_0x=b_0y$$

比较，则自然角频率由下式给出：

$$\omega_n^2=\frac{1}{LC}=\frac{1}{2.0\times20\times10^{-6}}$$

因此 $\omega_n=158\mathrm{Hz}$。与一般的二阶方程比较，又有：

$$\zeta^2=\frac{(R/L)^2}{4\times(1/LC)}=\frac{R^2C}{4L}=\frac{100^2\times20\times10^{-6}}{4\times2.0}$$

于是 $\zeta=0.16$。由于 ζ 小于 1，所以该系统欠阻尼。阻尼振荡频率 ω 为：

$$\omega=\omega_n\sqrt{1-\zeta^2}=158\sqrt{1-0.16^2}=156\mathrm{Hz}$$

因为系统欠阻尼，所以解与下式有相同的形式：

$$x=e^{-\zeta\omega_n t}(P\cos\omega t+Q\sin\omega t)+\frac{F}{k}$$

因此

$$v_C=e^{-0.16\times158t}(P\cos156t+Q\sin156t)+V$$

由于当 $t=0$ 时 $v_C=0$，则 $0=1(P+0)+V$。所以 $P=-V$。由于当 $t=0$ 时 $\frac{dv_C}{dt}=0$，所以对上式进行微分并等于 0，则给出：

$$\frac{dv_C}{dt}=e^{-\zeta\omega_n t}(-\omega P\sin\omega t+\omega Q\cos\omega t)-\zeta\omega_n e^{-\zeta\omega_n t}(P\cos\omega t+Q\sin\omega t)$$

这样 $0=1(0+\omega Q)-\zeta\omega_n(P+0)$，因此

$$Q=\frac{\zeta\omega_n P}{\omega}=-\frac{\zeta\omega_n V}{\omega}=-\frac{0.16\times158}{156}\approx-0.16V$$

所以微分方程的解为：

$$v_C=V-Ve^{-25.3t}(\cos156t+0.16\sin156t)$$

现考虑图 19-11 所示的系统。输入转矩 T 施加到一个圆盘上，其关于转轴中轴线的转动惯量为 I。该轴可以在圆盘一端自由旋转，但轴的远端固定。轴的旋转受扭转刚度反抗。输入转角为 θ_o 时产生反抗转矩 $k\theta_o$，其中 k 是常量。摩擦力也阻碍轴的旋转，并提供一个反抗转矩 $c\mathrm{d}\theta_o/\mathrm{d}t$，其中 c 是常量。假设我们要确定使该系统处于临界阻尼情况的条件。

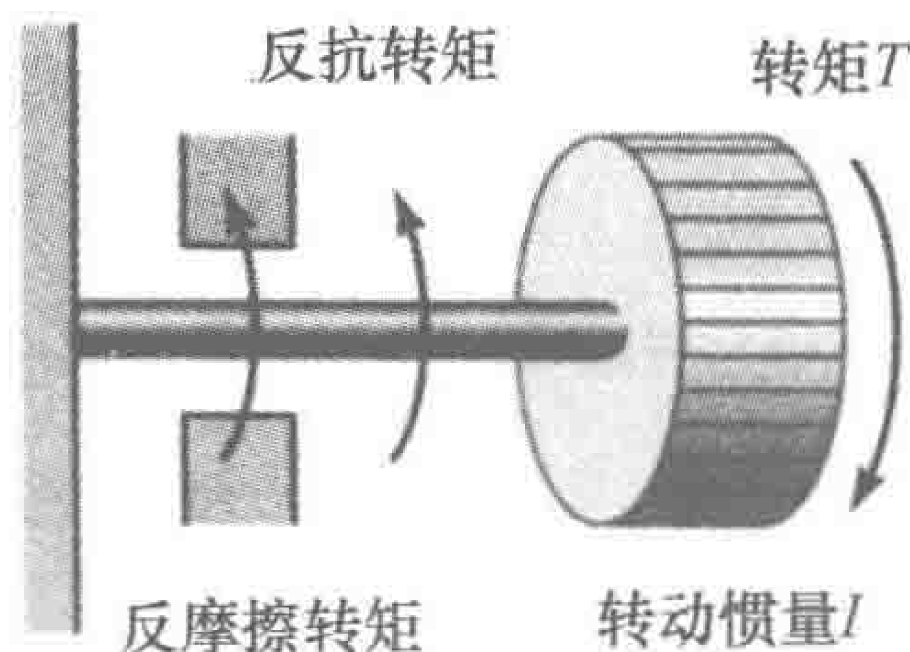


图 19-11　扭转系统

我们首先需要获得该系统的微分方程。净转矩为：

$$净转矩=T-c\frac{d\theta_0}{dt}-k\theta_0$$

462
~
463

净转矩等于 $I\frac{d^2\theta_0}{dt^2}$，故

$$I\frac{d^2\theta_0}{dt^2}=T-c\frac{d\theta_0}{dt}-k\theta_0$$

$$I\frac{d^2\theta_0}{dt^2}+c\frac{d\theta_0}{dt}+k\theta_0=T$$

当阻尼因子 ζ 等于 1 时，给出了临界阻尼的条件。将上述微分方程与二阶微分方程的一般形式进行比较，则

$$\zeta^2=\frac{a_1^2}{4a_2a_0}=\frac{c^2}{4Ik}$$

因此若使系统处于临界阻尼状态，必须使 $c=2\sqrt{Ik}$。

19.5　二阶系统的性能指标

图 19-12 给出了一个欠阻尼二阶系统对阶跃输入的典型响应曲线。针对这一响应给出了一些术语。

上升时间 t_r 是指输出响应 x 从 0 上升到稳态值 x_{SS} 所用的时间，它是用来表示系统对输入响应的速度。对于振荡响应来说，该时间是振荡完成 1/4 周期的时间，即 π/2。故，

$$\omega t_r=\frac{1}{2}\pi$$

有时上升时间还指输出响应从稳态值的某个指定比例（如 10%）到另外一个指定比例（如 90%）所用的时间。

464

峰值时间 t_P 是指响应从 0 上升到第一个峰值所用的时间。对于振荡响应来说，该时间是完成半个周期的时间，即 π。故，

$$\omega t_P=\pi$$

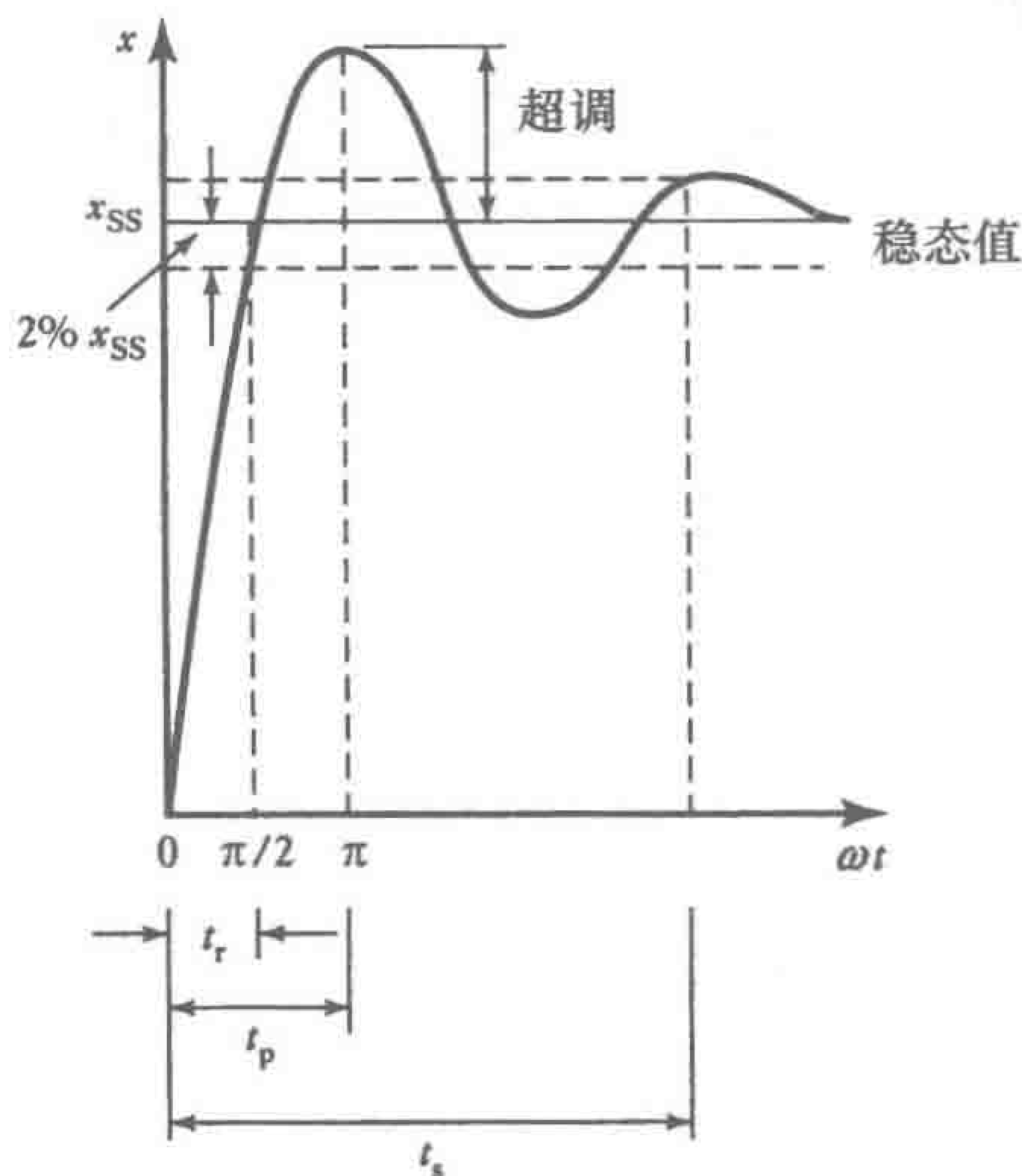


图 19-12　欠阻尼系统的阶跃响应

超调是指输出响应超过稳态值的最大量。因此，它是第一个峰值的振幅。超调常常被写为稳态值的百分比。对于系统的欠阻尼振荡，我们可写为：

$$x=e^{-\zeta\omega_n t}(P\cos\omega t+Q\sin\omega t)+稳态值$$

由于 $t=0$ 时 $x=0$，则 $0=1(P+0)+x_{SS}$，所以 $P=-x_{SS}$。在 $\omega t=\pi$ 时发生超调，故

$$x=e^{-\zeta\omega_n\pi/\omega}(P+0)+x_{SS}$$

超调值是峰值时间的输出响应与稳态值之差，因此

$$超调值=x_{SS}e^{-\zeta\omega_n\pi/\omega}$$

由于 $\omega=\omega_n\sqrt{1-\zeta^2}$，所以上式可写为：

$$超调值=x_{SS}\exp\left(\frac{-\zeta\omega_n\pi}{\omega_n\sqrt{1-\zeta^2}}\right)=x_{SS}\exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

以 x_{SS} 的百分比表示，有：

$$超调百分比=\exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)\times 100\%$$

表 19-2 给出了特定阻尼因子对应的超调百分比。

表 19-2　峰值超调百分比

阻尼因子	超调百分比	阻尼因子	超调百分比
0.2	52.7	0.6	9.5
0.4	25.4	0.8	1.5

振荡衰减有多快是用**衰减比**（subsidence ratio）或**衰减率**（decrement）来表示的。它是由第二次超调的振幅除以第一次超调的振幅得到。第一次超调发生在 $\omega t=\pi$ 时，第二次超调发生在 $\omega t=3\pi$ 时。于是

$$第一次超调值 = x_{SS}\exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

465

$$第二次超调值 = x_{SS}\exp\left(\frac{-3\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

故

$$衰减比 = \frac{第二次超调值}{第一次超调值} = \exp\left(\frac{-2\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

建立时间（settling time）t_s 是指振荡消失所用的时间。它是指振荡振幅下降并保持在某些规定的稳态值百分比（如 2%）所用的时间（见图 19-12）。这意味着振荡振幅应该小于 x_{SS} 的 2%。我们有

$$x = e^{-\zeta\omega_n t}(P\cos\omega t + Q\sin\omega t) + 稳态值$$

如前面推导，$P=-x_{SS}$。当 x 是一个最大值时，振荡振幅是（$x-x_{SS}$）。当 ωt 是 π 的倍数时，产生最大值，于是我们有 $\cos\omega t=-1$，$\sin\omega t=0$。建立时间 t_s 是指当振幅达到 x_{SS} 的 2%（即 $0.02x_{SS}$）时所用的时间，故

$$0.02x_{SS} = e^{-\zeta\omega_n t}(x_{SS}\times 1+0)$$

取对数，可得 $\ln 0.02=-\zeta\omega_n t_s$，由于 $\ln 0.02=-3.9$，或近似为 -4，所以

$$t_s = \frac{4}{\zeta\omega_n}$$

上式是指达到稳态值附近 2% 以内的建立时间。如果百分比是 5%，则方程变为：

$$t_s = \frac{3}{\zeta\omega_n}$$

由于完成一个周期的时间是 $1/f$，其中 f 是频率，又因为 $\omega=2\pi f$，所以完成一个周期的时间为 $2\pi/\omega$。在建立时间 t_s 内，发生振荡的次数是：

$$振荡数 = \frac{建立时间}{周期时间}$$

故对于稳态值的 2% 定义建立时间，有

$$振荡数 = \frac{4/\zeta\omega_n}{2\pi/\omega}$$

由于 $\omega=\omega_n\sqrt{1-\zeta^2}$，所以

$$振荡数 = \frac{2\omega_n\sqrt{1-\zeta^2}}{\pi\zeta\omega_n} = \frac{2}{\pi}\sqrt{\frac{1}{\zeta^2}-1}$$

为了说明上式，考虑自然角频率为 2.0Hz，阻尼频率为 1.8Hz 的二阶系统。由于 $\omega=\omega_n\sqrt{1-\zeta^2}$，所以阻尼因子可由下式给出

$$1.8 = 2.0\sqrt{1-\zeta^2}$$

并且$\zeta=0.44$。由于$\omega t_r=\pi/2$，则100%的上升时间由下式给出

466

$$t_r = \frac{\pi}{2\times 1.8} = 0.87\text{s}$$

超调的百分比由下式给出

$$超调百分比 = \exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)\times 100\%$$

$$= \exp\left(\frac{-0.44\pi}{\sqrt{1-0.44^2}}\right)\times 100\%$$

故超调的百分比是21%。2%建立时间由下式给出

$$t_s = \frac{4}{\zeta\omega_n} = \frac{4}{0.44\times 2.0} = 4.5\text{s}$$

在2%建立时间内的振荡次数为

$$振荡数 = \frac{2}{\pi}\sqrt{\frac{1}{\zeta^2}-1} = \frac{2}{\pi}\sqrt{\frac{1}{0.44^2}-1} = 1.3$$

19.6 系统辨识

第17、18章中系统模型的建立，是以假设它们具有简单的结构为前提。另一种可选的建立真实系统模型的方法是：通过测试确定某些输入的响应，例如阶跃输入，然后找到与这个响应匹配的模型。确定这个数学模型的过程称为**系统辨识**。因此，如果获得阶跃输入的响应形式如图19-5所示，则可假设它是一阶系统，并可从响应曲线来确定时间常量。例如，假设响应用了1.5s的时间达到了最终信号高度的0.63倍，且最终信号高度是阶跃输入的5倍。表19-1表示了1.5s的时间常量，故描述模型的微分方程为：

$$1.5\frac{\mathrm{d}x}{\mathrm{d}t} + x = 5y$$

欠阻尼二阶系统将给出一个如图19-12所示的阶跃输出响应。阻尼比可由对第一和第二个超调的测量与它们二者之比（即衰减比）给出。自然频率可由连续超调之间的时间来决定。于是，我们可用这些值来确定二阶微分方程中的常量。

本章小结

自然响应是指当系统没有使系统变量改变的输入，而只是发生自然变化时的系统响应。系统的**受迫响应**是当有输入到该系统并使输出发生改变的响应。一个没有受迫输入的一阶系统有如下形式的微分方程：

$$a_1\frac{\mathrm{d}x}{\mathrm{d}t} + a_0 x = 0$$

467

并且该方程有解$x=\mathrm{e}^{-a_0t/a_1}$。

当有**受迫函数**时，该微分方程有如下的形式：

$$a_1\frac{\mathrm{d}x}{\mathrm{d}t} + a_0 x = b_0 y$$

其解为：

$$x = 稳态值 \times (1 - e^{-a_0 t/a_1})$$

时间常量 τ 是指输出上升到稳态值的 0.63 倍时所用的时间，为 a_1/a_0。

一个没有受迫输入的二阶系统有如下形式的微分方程：

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0$$

自然角频率由 ${\omega_n}^2 = k/m$ 给出，阻尼常量由 $\zeta^2 = c^2/4mk$ 给出。当 $\zeta > 1$ 时，系统**过阻尼**，且 x_n 的一般解为：

$$x_n = Ae^{s_1 t} + Be^{s_2 t} \quad 和 \quad s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

当 $\zeta = 1$ 时，系统处于**临界阻尼**状态，则

$$x_n = (At + B)e^{-\omega_n t}$$

当 $\zeta < 1$ 时，系统处于**欠阻尼**状态，则

$$x_n = e^{-\zeta\omega_n t}(P\cos\omega t + Q\sin\omega t)$$

当有受迫输入 F 时，二阶微分方程变为：

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = F$$

对于过阻尼系统：

$$x = Ae^{s_1 t} + Be^{s_2 t} + \frac{F}{k}$$

对于临界阻尼系统：

$$x = (At + B)e^{-\omega_n t} + \frac{F}{k}$$

对于欠阻尼系统：

$$x = e^{-\zeta\omega_n t}(P\cos\omega t + Q\sin\omega t) + \frac{F}{k}$$

上升时间 t_r 是响应 x 从 0 上升到稳态值 x_{SS} 所用的时间，它是系统对输入响应的快速性的一个指标，由 $\omega t_r = \pi/2$ 给出。**峰值时间** t_p 是响应从 0 上升到第一个峰值所用的时间，由 $\omega t_p = \pi$ 给出。**超调**是指响应超过稳态值的最大量，即

$$超调值 = x_{SS}\exp\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

468

衰减比或**衰减率**是指第二个超调的振幅除以第一个超调的振幅得到的比值，即

$$衰减比 = \exp\left(\frac{-2\zeta\pi}{\sqrt{1-\zeta^2}}\right)$$

建立时间 t_s 是指系统响应下降并保持在稳态值的某些指定的百分比（如 2%）内所用的时间，由下式给出：

$$t_s = \frac{4}{\zeta\omega_n}$$

习题

19.1　一个一阶系统的时间常量是 4s，稳态增益是 6。该系统微分方程的形式是什么？

19.2　玻璃水银温度计的时间常量是 10s。如果突然从 20℃ 插入 80℃ 的热水中，在（a）10s；（b）20s 后温度计的指示温度是多少？

19.3 一个电路由电阻 R 和电感 L 串联组成。当 $t=0$ 时刻有阶跃输入电压 V 时，系统的微分方程为：

$$\frac{\mathrm{d}i}{\mathrm{d}t}+\frac{R}{L}i=\frac{V}{L}$$

求：

(a) 微分方程的解。

(b) 时间常量。

(c) 稳态电流 i。

19.4 描述一个阻尼因子分别是（a）0；（b）0.5；（c）1.0 和（d）1.5 的二阶系统对阶跃输入的输出随时间的变换形式。

19.5 一个 RLC 电路当有阶跃输入 V 时，电容 C 两端的电压 v_C 随着时间发生变化，其微分方程描述为：

$$\frac{\mathrm{d}^2v_C}{\mathrm{d}t^2}+10\frac{\mathrm{d}v_C}{\mathrm{d}t}+16v_C=16V$$

求：

(a) 自然频率。

(b) 阻尼因子。

(c) 当 $t=0$ 时 $v_C=0$，当 $t=0$ 时 $\frac{\mathrm{d}v_C}{\mathrm{d}t}=0$ 对应的方程解。

19.6 一个系统当有阶跃输入 y 时其输出 x 随着时间发生变化，可被描述为：

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+10\frac{\mathrm{d}x}{\mathrm{d}t}+25x=50y$$

求：

(a) 自然频率。

(b) 阻尼系数。

(c) 当 $t=0$ 时 $x=0$，当 $t=0$ 时 $\frac{\mathrm{d}x}{\mathrm{d}t}=-2$，且存在一个 3 个单位的阶跃输入，求对应的方程解。

19.7 一个加速度计（即用于测量加速度的仪器）的无阻尼角频率（或自然角频率）为 100Hz，阻尼因子为 0.6。求：

(a) 最大的超调百分比。

469 (b) 当加速度突然变化时的上升时间。

19.8 一个系统给出了如下有阶跃输入 y 的微分方程：

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+5\frac{\mathrm{d}x}{\mathrm{d}t}+16x=16y$$

求：

(a) 自然频率。

(b) 阻尼因子。

(c) 阻尼角频率。

(d) 上升时间。

(e) 最大的超调的百分比。

(f) 达到 0.2% 稳定的时间。

19.9 当 10V 的电压突然加到一个动圈式电压表上时，仪表指针在最终稳定到读数 10V 前上升到 11V。求：

(a) 阻尼因子。

(b) 当指针处于稳态值 0.2% 的范围前，指针的振荡数。

19.10 一个二阶系统由如下的微分方程描述：

$$\frac{d^2x}{dt^2} + c\frac{dx}{dt} + 4x = F$$

如果超调的百分比小于 9.5%，则阻尼常量 c 将是多少？

19.11 观察阻尼系统的振荡，输入响应显示在第二个周期内的最大超调是第一个超调的 75%。请问系统的阻尼因子是多少？

19.12 发现一个二阶系统在第一次超调和第二次超调之间的时间是 1.6s。请问系统的自然频率是多少？

470

第 20 章　系统传递函数

本章目的

通过本章的学习，读者将能够：

- 定义传递函数及通过一阶和二阶系统的微分方程确定传递函数。
- 确定具有反馈系统的传递函数。
- 应用拉普拉斯变换，确定一阶和二阶系统对于简单输入的响应。
- 确定极点位置对于系统响应的影响。

20.1　传递函数

对于一个放大器系统，一般会涉及放大器的**增益**。它显示了输出信号相比于输入信号的放大量。它能使输出信号由特定的输入所决定。例如，对于一个电压增益为 10 的放大器，如果输入电压是 2mV，输出将为 20mV；又如果输入电压是 1V，输出将为 10V。增益表示了模块中输入和输出间的数学关系。我们指出当信号是时域时，即时间的函数，则可把它写成 $f(t)$。这样，对于输入 $y(t)$ 和输出 $x(t)$（见图 20-1a），有下式

$$\text{增益} = \frac{\text{输出}}{\text{输入}} = \frac{x(t)}{y(t)}$$

$y(t)$ → 增益 G → $x(t)$

a）时域

$Y(s)$ → 传递函数 $G(s)$ → $X(s)$

b）s域

图 20-1　框图

但是，对于许多系统输入和输出的关系是微分方程的形式，所以仅用一个像增益 10 这样简单的数字来描述一个函数是不可能的。因为输入和输出的关系是微分方程而不是简单的代数方程，所以不能简单地把输出除以输入。但是可以运用称为**拉普拉斯变换**（Laplace transform）的方法把微分方程转化成代数方程。微分方程描述系统随时间的变化，可以用拉普拉斯变换的方法使其变换成简单的与时间无关的代数方程，并可以进行一般的代数运算。我们讨论将时域的行为转化为 s 域。当一个函数被写成 s 域形式时，因为它是 s 的函数，所以可写成 $F(s)$。一般来说，用大写字母 F 来代表拉普拉斯变换，用小写字母 f 代表随时间变化的函数 $f(t)$。

因此把输入和输出的关系定义成一个**传递函数**。它表示输入的拉普拉斯变换与输出的拉普拉斯变换之间的关系。假设输入到线性系统的输入信号的拉普拉斯形式是 $Y(s)$，输出信号的拉普拉斯形式是 $X(s)$。系统的传递函数 $G(s)$ 定义为

$$\text{传递函数} = \text{输出的拉普拉斯变换} / \text{输入的拉普拉斯变换}$$

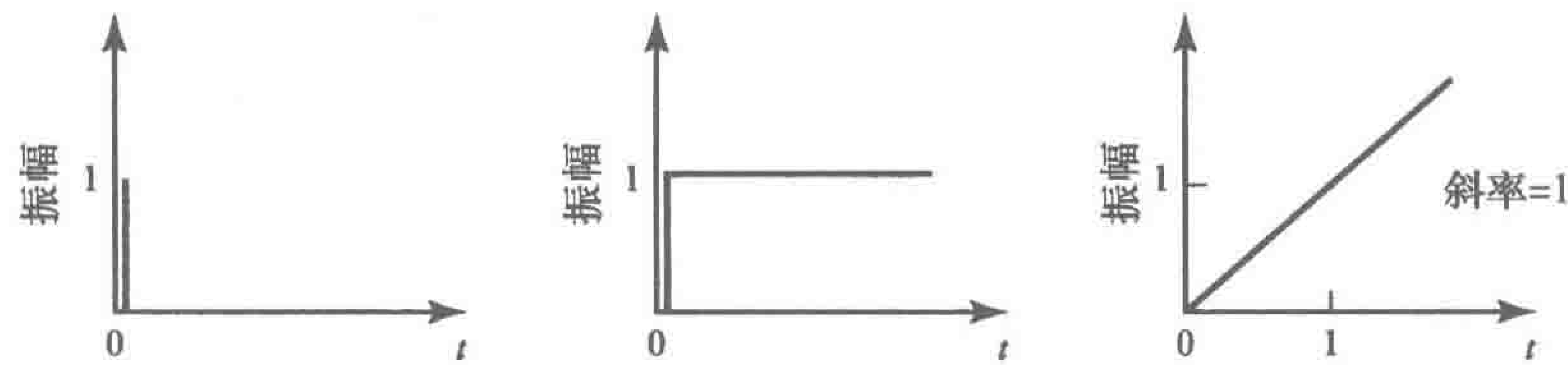
$$G(s) = \frac{X(s)}{Y(s)}$$

其中，所有初始条件为 0，即假设零输入时得到零输出，输入随时间的变化率是 0 时，输出随时间的变化率也是 0。这样输出变换是 $X(s) = G(s)Y(s)$，即输入变换和传递函数的乘积。若

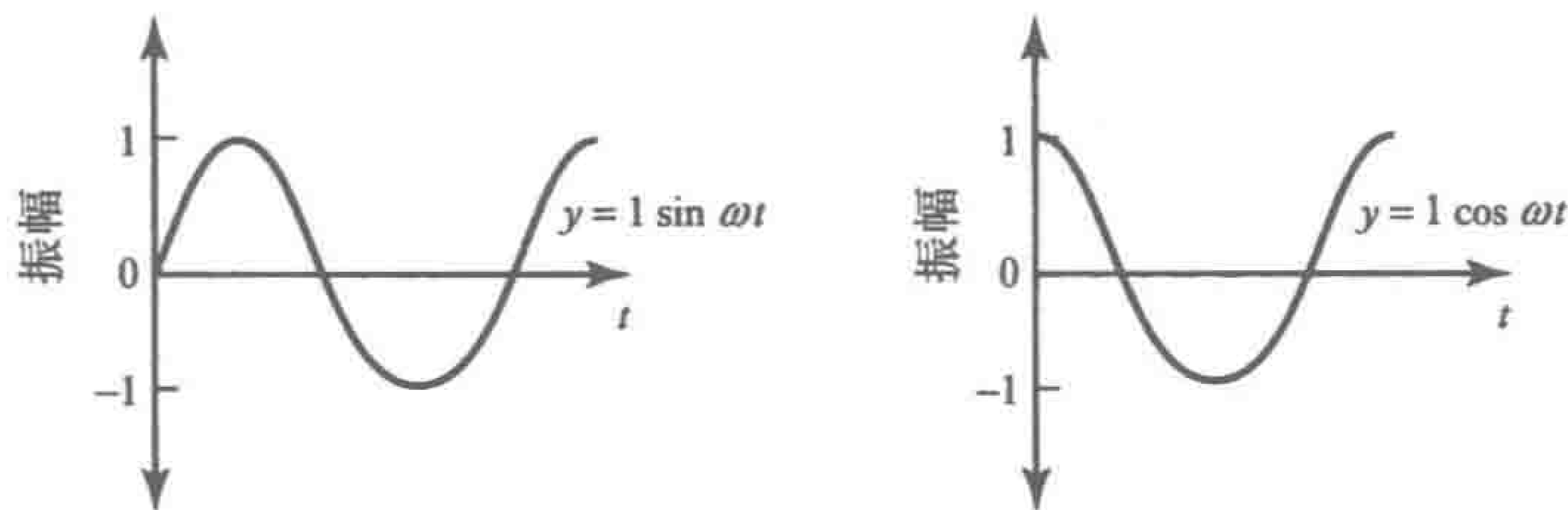
用框图（见图 20-1b）表示 s 域中的系统，则框中的函数 $G(s)$ 是一个把输入 $Y(s)$ 变换成输出 $X(s)$ 的函数。

20.1.1　拉普拉斯变换

为了得到包含时间函数的微分方程的拉普拉斯变换，可以运用表格和一些基本准则（附录 A 包含这样一个表格，并且给出规则的细节）。图 20-2 是常见输入形式的基本变换。



0时刻的单位脉冲输入的变换为1　0时刻的单位阶跃输入的变换为 $1/s$　0时刻的单位斜坡输入的变换为 $1/s^2$



单位振幅正弦波的变换为 $\omega/(s^2+\omega^2)$　单位振幅余弦波的变换为 $s/(s^2+\omega^2)$

图 20-2　输入的拉普拉斯变换

471 ~ 472

以下是拉普拉斯变换的一些基本准则：

1）如果一个时间函数乘以一个常量，则其拉普拉斯变换后也乘以相同的常量，即

$$af(t) \text{ 的拉普拉斯变换是 } aF(s)$$

例如，6V 阶跃输入到电路系统的拉普拉斯变换是单位阶跃输入的拉普拉斯变换的 6 倍，即 6s。

2）如果方程是两个独立的时间函数的和，则该方程的拉普拉斯变换是两个独立的拉普拉斯变换后的和，即

$$f(t) + g(t) \text{ 的变换是 } F(s) + G(s)$$

3）函数一阶导数的拉普拉斯变换是

$$\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0)$$

其中，$f(0)$ 是 $f(t)$ 在 $t=0$ 时的初始值。但是，处理传递函数时我们一般认为所有初始条件都为 0。

4）二阶导数的拉普拉斯变换是

$$\mathcal{L}\left\{\frac{d^2}{dt^2}f(t)\right\} = s^2F(s) - sf(0) - \frac{d}{dt}f(0)$$

其中，$df(0)/dt$ 是 $df(t)/dt$ 在 $t=0$ 时的初始值。但是，处理传递函数时我们一般认为所有初始条件都为 0。

5）函数积分的拉普拉斯变换是

$$\mathcal{L}\left\{\int_0^t f(t)\,dt\right\} = \frac{1}{s}F(s)$$

这样，当所有初始条件是 0 时，为求得微分或积分函数的拉普拉斯变换，我们可以：

用 $F(s)$ 代替时间函数 $f(t)$；

用 $sF(s)$ 代替函数一阶导数 $df(t)/dt$；

用 $s^2F(s)$ 代替二阶导数 $d^2f(t)/dt^2$；

用 $F(s)/s$ 代替 $\int f(t)dt$。

在 s 域进行代数运算后，通过应用拉普拉斯逆变换的表格，可以把输出转换回时域，即找到对应 s 域结果的时域函数。通常，变换时需要把方程重新表示成表格中给出的形式。以下是一些有用的逆变换函数，更多的细节可以见附录 A。

473

	拉普拉斯变换	时间函数
1.	$\frac{1}{s+a}$	e^{-at}
2.	$\frac{a}{s(s+a)}$	$(1-e^{-at})$
3.	$\frac{b-a}{(s+a)(s+b)}$	$e^{-at}-e^{-bt}$
4.	$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}$
5.	$\frac{a}{s^2(s+a)}$	$t-\frac{1-e^{-at}}{a}$

下面将讲述变换在一阶和二阶系统中的应用。

20.2　一阶系统

考虑一个系统，它的输入和输出的关系可用一阶微分方程表示。一阶系统的微分方程有如下的形式：

$$a_1\frac{dx}{dt}+a_0x=b_0y$$

其中，a_1，a_2，b_0 是常量，y 是输入，x 是输出，输入和输出都是时间的函数。所有的初始条件是 0，则其拉普拉斯变换是

$$a_1sX(s)+a_0X(s)=b_0Y(s)$$

因此可以把传递函数 $G(s)$ 写成

$$G(s)=\frac{X(s)}{Y(s)}=\frac{b_0}{a_1s+a_0}$$

它可以重新整理为

$$G(s)=\frac{b_0/a_0}{(a_1/a_0)s+1}=\frac{G}{\tau s+1}$$

其中，G 是稳态条件下的系统增益，即没有 dx/dt 项。a_1/a_0 是系统的时间常量 τ（见 19.3.4 节）。

20.2.1　具有阶跃输入的一阶系统

当一个单位阶跃信号输入到一阶系统中时，$Y(s)=1/s$，输出变换 $X(s)$ 是

$$X(s)=G(s)Y(s)=\frac{G}{s(\tau s+1)}=G\frac{(1/\tau)}{s(s+1/\tau)}$$

因此，因为我们已经有 $a/s(s+a)$ 的拉普拉斯变换，所以运用 20.1.1 节项目 2 的逆变换可得

474

$$x=G(1-e^{-t/\tau})$$

20.2.2　一阶系统的例子

以下的例子展示了如何基于以上的知识点，分析具有阶跃输入的一阶系统传递函数和它的行为。

1）分析一个由电阻 R 和电容 C 串联组成的电路。电路的输入是 v，输出是电容两端的电势差 v_C。输入和输出的微分方程是：

$$v = RC\frac{dv_C}{dt} + v_C$$

下面确定传递函数。

进行拉普拉斯变换，所有的初始条件是 0，则

$$V(s) = RCsV_C(s) + V_C(s)$$

故传递函数是

$$G(s) = \frac{V_C(s)}{V(s)} = \frac{1}{RCs + 1}$$

2）分析一个热电偶，其电压输出 V 和温度输入的传递函数是

$$G(s) = \frac{30 \times 10^{-6}}{10s + 1}\text{V/℃}$$

试确定 100℃的阶跃输入下，系统的响应和上升到稳态值的 95% 所需要的时间。

因为输出的拉普拉斯变换等于传递函数和输入的拉普拉斯变换的乘积，则

$$V(s) = G(s) \times \text{input}(t)$$

100℃的阶跃输入相当于热电偶的温度突变为 100℃，其拉普拉斯变换为 100/s。则

$$V(s) = \frac{30 \times 10^{-6}}{10s + 1} \times \frac{100}{s} = \frac{30 \times 10^{-4}}{10s(s + 0.1)}$$

$$= 30 \times 10^{-4}\frac{0.1}{s(s + 0.1)}$$

分数部分是 $a/s(s+a)$ 的形式，故逆变换为

$$V = 30 \times 10^{-4}(1 - e^{-0.1t})\text{V}$$

终值，即稳态值，是 $t\to\infty$ 时的值，所以指数项为 0。终值是 30×10^{-4}V。上升到稳态值的 95% 的时间可以由下式给出

$$0.95 \times 30 \times 10^{-4} = 30 \times 10^{-4}(1 - e^{-0.1t})$$

这样，$0.05 = e^{-0.1t}$，且 $\ln(0.05) = 0.1t$。于是，时间为 30s。

475

3）考虑将 $5t$℃/s 的斜坡输入到上述热电偶系统中，即温度每秒上升 5℃。试分析热电偶输出电压随时间的变化及 12s 后的电压。

该斜坡信号的拉普拉斯变换为 $5/s^2$。于是：

$$V(s) = \frac{30 \times 10^{-6}}{10s + 1} \times \frac{5}{s^2} = 150 \times 10^{-6}\frac{0.1}{s^2(s + 0.1)}$$

运用 20.1.1 节列出的项目 5 进行逆变换，则有：

$$V = 150 \times 10^{-6}\left(t - \frac{1 - e^{-0.1t}}{0.1}\right)$$

12s 后电压是 $V = 7.5 \times 10^{-4}$V。

4）分析 100℃的脉冲输入，即热电偶的温度短暂地升至 100℃。试分析热电偶输出电压随时间的变化及 2s 后的电压。

脉冲信号的拉普拉斯变换是 100，故

$$V(s) = \frac{30 \times 10^{-6}}{10s + 1} \times 100 = 3 \times 10^{-4}\frac{1}{s + 0.1}$$

因此，$V = 3 \times 10^{-4}e^{-0.1t}$V。2s 后，热电偶的电压为 $V \approx 2.46 \times 10^{-4}$V。

20.3　二阶系统

对于一个二阶系统，输入 y 和输出 x 之间的关系可以描述成以下形式的微分方程

$$a_2\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+a_1\frac{\mathrm{d}x}{\mathrm{d}t}+a_0x=b_0y$$

其中，a_2，a_1，a_0 和 b_0 是常量。所有的初始条件是 0，该方程的拉普拉斯变换是

$$a_2s^2X(s)+a_1sX(s)+a_0X(s)=b_0Y(s)$$

故

$$G(s)=\frac{X(s)}{Y(s)}=\frac{b_0}{a_2s^2+a_1s+a_0}$$

二阶系统微分方程的另一种表示方法是

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+2\zeta\omega_{\mathrm{n}}\frac{\mathrm{d}x}{\mathrm{d}t}+\omega_{\mathrm{n}}^2x=b_0\omega_{\mathrm{n}}^2y$$

其中，ω_{n} 是系统的自然角频率，ζ 是阻尼比。方程的拉普拉斯变换是

$$G(s)=\frac{X(s)}{Y(s)}=\frac{b_0\omega_{\mathrm{n}}^2}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2}$$

476　以上是二阶系统传递函数的常用表达形式。

20.3.1　具有阶跃输入的二阶系统

当把单位阶跃信号输入到一个二阶系统时，即 $Y(s)=1/s$，则输出变换为

$$X(s)=G(s)Y(s)=\frac{b_0\omega_{\mathrm{n}}^2}{s(s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2)}$$

它可变换成

$$X(s)=\frac{b_0\omega_{\mathrm{n}}^2}{s(s+p_1)(s+p_2)}$$

其中，p_1 和 p_2 是以下方程的根：

$$s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2=0$$

故应用二次方程的求根公式为：

$$p=\frac{-2\zeta\omega_{\mathrm{n}}\pm\sqrt{4\zeta^2\omega_{\mathrm{n}}^2-4\omega_{\mathrm{n}}^2}}{2}$$

则两个根 p_1 和 p_2 分别是

$$p_1=-\zeta\omega_{\mathrm{n}}+\omega_{\mathrm{n}}\sqrt{\zeta^2-1},\quad p_2=-\zeta\omega_{\mathrm{n}}-\omega_{\mathrm{n}}\sqrt{\zeta^2-1}$$

当 $\zeta>1$ 时，平方根项是实数，系统是过阻尼系统。为了求逆变换，可以用部分分式法（见附录 A）把表达式分解成一系列简单的分式，或用附录 A 中变换表中的项目 14。任何一种方法都可得到如下的结果

$$x=\frac{b_0\omega_{\mathrm{n}}^2}{p_1p_2}\left[1-\frac{p_2}{p_2-p_1}\mathrm{e}^{-p_1t}+\frac{p_1}{p_2-p_1}\mathrm{e}^{-p_2t}\right]$$

当 $\zeta=1$ 时，平方根项是 0，$p_1=p_2=-\omega_{\mathrm{n}}$。系统是临界阻尼系统。方程变为：

$$X(s)=\frac{b_0\omega_{\mathrm{n}}^2}{s(s+\omega_{\mathrm{n}})^2}$$

方程用部分分式法（见附录 A）展开，得到

$$Y(s)=b_0\left[\frac{1}{s}-\frac{1}{s+\omega_{\mathrm{n}}}-\frac{\omega_{\mathrm{n}}}{(s+\omega_{\mathrm{n}})^2}\right]$$

故

$$x = b_0[1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}]$$

当 $\zeta<1$ 时，则

$$x = b_0\left[1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin(\omega_n \sqrt{(1-\zeta^2)}\, t + \phi)\right]$$

其中，$\cos\phi=\zeta$。这是欠阻尼振荡。

477

20.3.2　二阶系统的例子

以下的例子说明了以上内容。

1）系统传递函数如下，单位阶跃输入和系统的阻尼状态是什么？

$$G(s) = \frac{1}{s^2+8s+16}$$

单位阶跃输入 $Y(s)=1/s$，输出变换是

$$X(s) = G(s)Y(s) = \frac{1}{s(s^2+8s+16)} = \frac{1}{s(s+4)(s+4)}$$

因此，$s^2+8s+16$ 的根是 $p_1=p_2=-4$。两根都是实数且相等，故系统是临界阻尼系统。

2）机器人臂有如下的传递函数

$$G(s) = \frac{K}{(s+3)^2}$$

则单位斜坡信号输入下的输出是什么？

输出的拉普拉斯变换 $X(s)$ 是

$$X(s) = G(s)Y(s) = \frac{K}{(s+3)^2}\times\frac{1}{s^2}$$

运用部分分式法展开（见附录 A），得到

$$X(s) = \frac{K}{9s^2} - \frac{2K}{9(s+3)} + \frac{K}{9(s+3)^2}$$

因此，进行逆变换后有

$$x = \frac{1}{9}Kt - \frac{2}{9}Ke^{-3t} + \frac{1}{9}Kte^{-3t}$$

20.4　串联的系统

如果系统是由一系列子系统串联而成的，如图 20-3 所示，则传递函数 $G(s)$ 是

$$G(s) = \frac{X(s)}{Y(s)} = \frac{X_1(s)}{Y(s)}\times\frac{X_2(s)}{X_1(s)}\times\frac{X(s)}{X_2(s)}$$
$$= G_1(s)\times G_2(s)\times G_3(s)$$

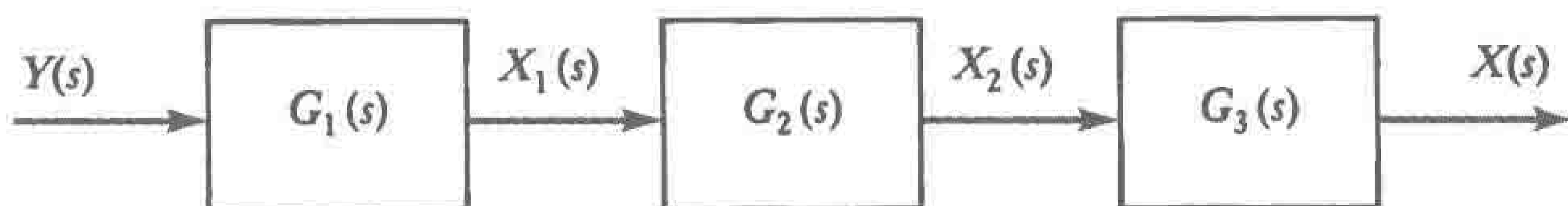


图 20-3　串联系统

串联系统的传递函数是各个子系统传递函数的乘积。

478

20.4.1　串联系统的例子

以下例子说明串联的系统。假设当子系统连接在一起时，各个模块之间没有相互影响，传递函数也不会变化。例如，当子系统相互影响、相互加载时，电路就会出现问题。

1）各个子模块的传递函数是10，$2/s$，$4/(s+3)$，将这三个元素串联起来组成的系统的传递函数是什么？

运用上述推导的公式，有

$$G(s)=10\times\frac{2}{s}\times\frac{4}{s+3}=\frac{80}{s(s+3)}$$

2）一个磁场控制的直流电动机由三个串联的子系统组成，分别是励磁电路、电枢绕组和负载。图20-4描述了子系统的布置和传递函数。试求系统的传递函数。

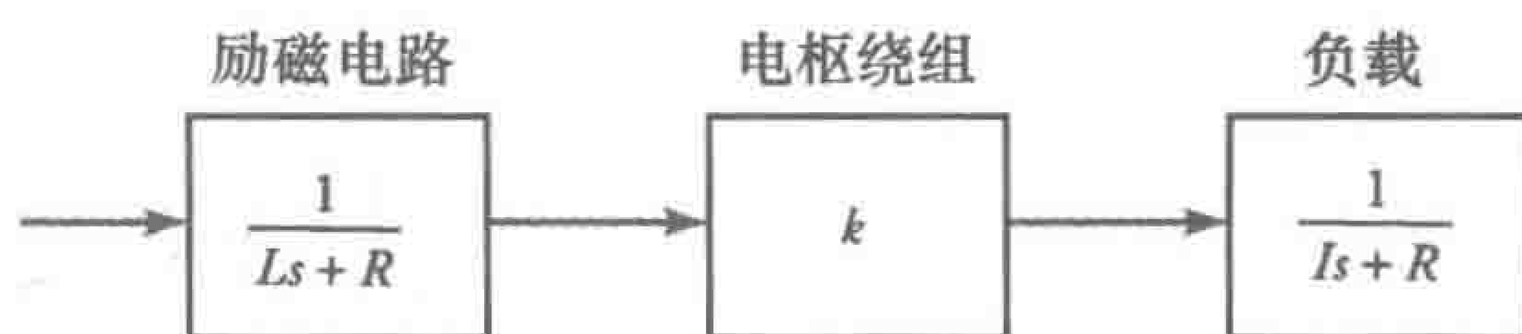


图20-4　磁场控制的直流电动机

串联系统的传递函数是串联模块的传递函数的乘积。

$$G(s)=\frac{1}{Ls+R}\times k\times\frac{1}{Is+c}=\frac{k}{(Ls+R)(Is+c)}$$

20.5　带反馈环的系统

图20-5表示了一个带有**负反馈**的简单系统。对于具有负反馈的系统，输入和反馈信号在求和点相减。**前向通道**是指图中有传递函数 $G(s)$ 的通道，而**反馈通道**是具有 $H(s)$ 的通道。整个系统称为**闭环系统**。

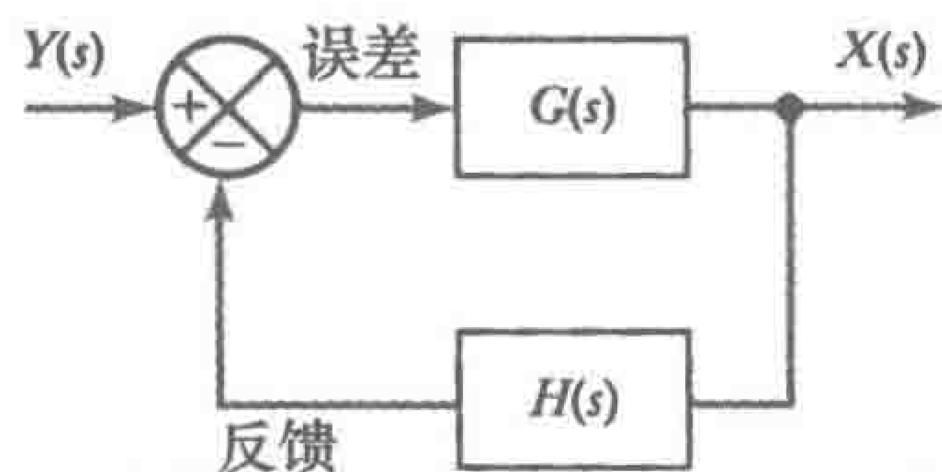


图20-5　负反馈系统

对于负反馈系统，具有前向通道传递函数 $G(s)$ 的子系统的输入是 $Y(s)$ 减去反馈信号。反馈环的传递函数是 $H(s)$，它的输入是 $X(s)$，因此反馈信号是 $H(s)X(s)$。$G(s)$ 的输入是 $Y(s)-H(s)X(s)$，输出是 $X(s)$，则：

$$G(s)=\frac{X(s)}{Y(s)-H(s)X(s)}$$

重新整理后，得出

$$\frac{X(s)}{Y(s)}=\frac{G(s)}{1+G(s)H(s)}$$

故整个负反馈系统 $T(s)$ 的传递函数是

479

$$T(s)=\frac{X(s)}{Y(s)}=\frac{G(s)}{1+G(s)H(s)}$$

20.5.1　负反馈系统的例子

以下的例子说明了负反馈系统。

1）前向通道的传递函数是 $2/(s+1)$，负反馈通道的传递函数是 $5s$，则闭环系统的整体传递函数是什么？

运用上述的公式

$$T(s)=\frac{G(s)}{1+G(s)H(s)}=\frac{2/(s+1)}{1+[2/(s+1)]5s}=\frac{2}{11s+1}$$

2）分析一个电枢控制的直流电动机（见图20-6）。前向通道由三个元素组成：电枢电路传递函数是 $1/(Ls+R)$ 的电枢电路、传递函数是 k 的电枢绕组、传递函数是 $1/(Is+c)$ 的负载。负反馈通道传递函数是 K。试求系统的传递函数。

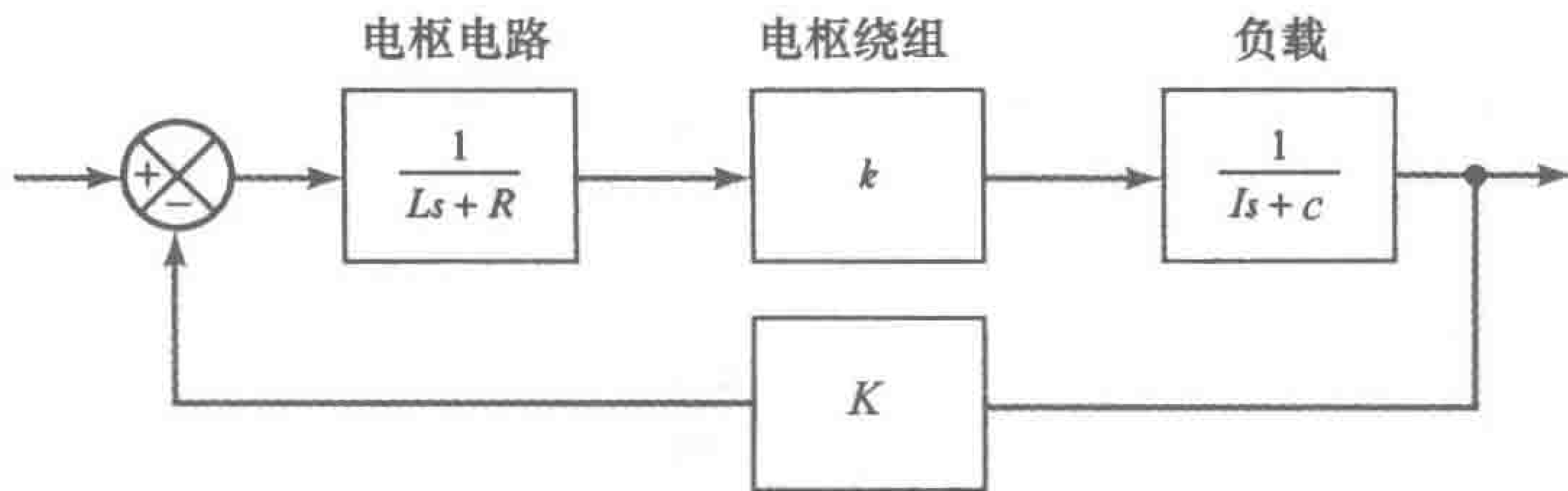


图 20-6 电枢控制的直流电动机

串联元件的前向通道传递函数是串联元件的传递函数的乘积，即

$$G(s)=\frac{1}{Ls+R}\times k\times\frac{1}{Is+c}=\frac{k}{(Ls+R)(Is+c)}$$

负反馈通道的传递函数是 K。故系统的传递函数是

$$T(s)=\frac{G(s)}{1+G(s)H(s)}=\frac{\dfrac{k}{(Ls+R)(Is+c)}}{1+\dfrac{kK}{(Ls+R)(Is+c)}}$$

$$=\frac{k}{(Ls+R)(Is+c)+kK}$$

20.6 极点位置对瞬态响应的影响

当给系统一个输入时，系统有随时间逐渐衰减的瞬态响应，并达到稳态条件，我们定义这样的系统是**稳定的**。如果瞬态响应不随时间衰减反而增强，则该系统称为**不稳定的**，因此该系统永远也达不到稳态条件。

480

考虑单位脉冲输入到传递函数为 $G(s)=1/(s+1)$ 的一阶系统。系统输出 $X(s)$ 是

$$X(s)=\frac{1}{s+1}\times 1$$

因此 $x=e^{-t}$。随着时间 t 的增加，输出逐渐衰减，最终变为 0。现在考虑单位脉冲输入到传递函数 $G(s)=1/(s-1)$ 的一阶系统。系统输出 $X(s)$ 是

$$X(s)=\frac{1}{s-1}\times 1$$

因此 $x=e^{t}$。随着时间 t 的增加，输出变大。因此，输入到系统的瞬时脉冲会导致一个递增的输出。该系统是不稳定的。

对于一个传递函数，使其为无穷大的 s 值称为它的**极点**。它们是特征方程的根。对于 $G(s)=1/(s+1)$，极点 $s=-1$。对于 $G(s)=1/(s-1)$，极点 $s=+1$。因此，对于一阶系统，如果极点是负数，则系统是稳定的；如果极点是正数，则系统是不稳定的（见图 20-7）。

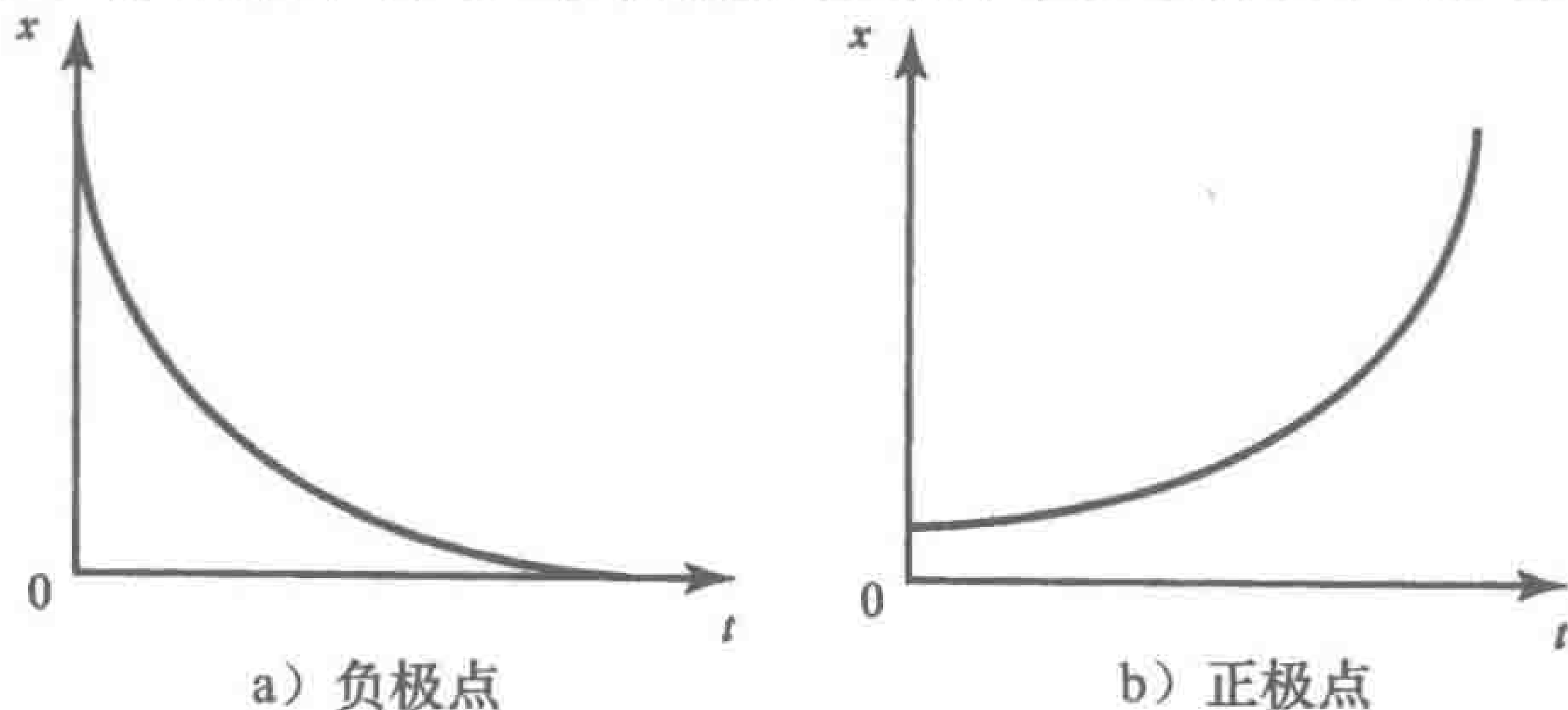


图 20-7 一阶系统

对于二阶系统，传递函数是

$$G(s)=\frac{b_0\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

当输入是单位脉冲输入时：

$$X(s)=\frac{b_0\omega_n^2}{(s+p_1)(s+p_2)}$$

其中，p_1 和 p_2 是以下方程的根。

$$s^2+2\zeta\omega_n s+\omega_n^2=0$$

应用二次方程的求根公式，

$$p=\frac{-2\zeta\omega_n\pm\sqrt{4\zeta^2\omega_n^2-4\omega_n^2}}{2}=-\zeta\omega_n\pm\omega_n\sqrt{\zeta^2-1}$$

481 根据阻尼因子的值，平方根是实数或者虚数。当输出包含虚数项时，输出是振荡的。例如，假设一个二阶系统，它的传递函数是

$$G(s)=\frac{1}{[s-(-2+j1)][s-(-2-j1)]}$$

即 $p=-2\pm j1$。当输入是单位脉冲时，输出是 $e^{-2t}\sin t$。振荡的振幅是 e^{-2t}，随着时间的推移慢慢衰减，因此脉冲的响应是逐渐衰减的振荡（见图 20-8a）。系统是稳定的。

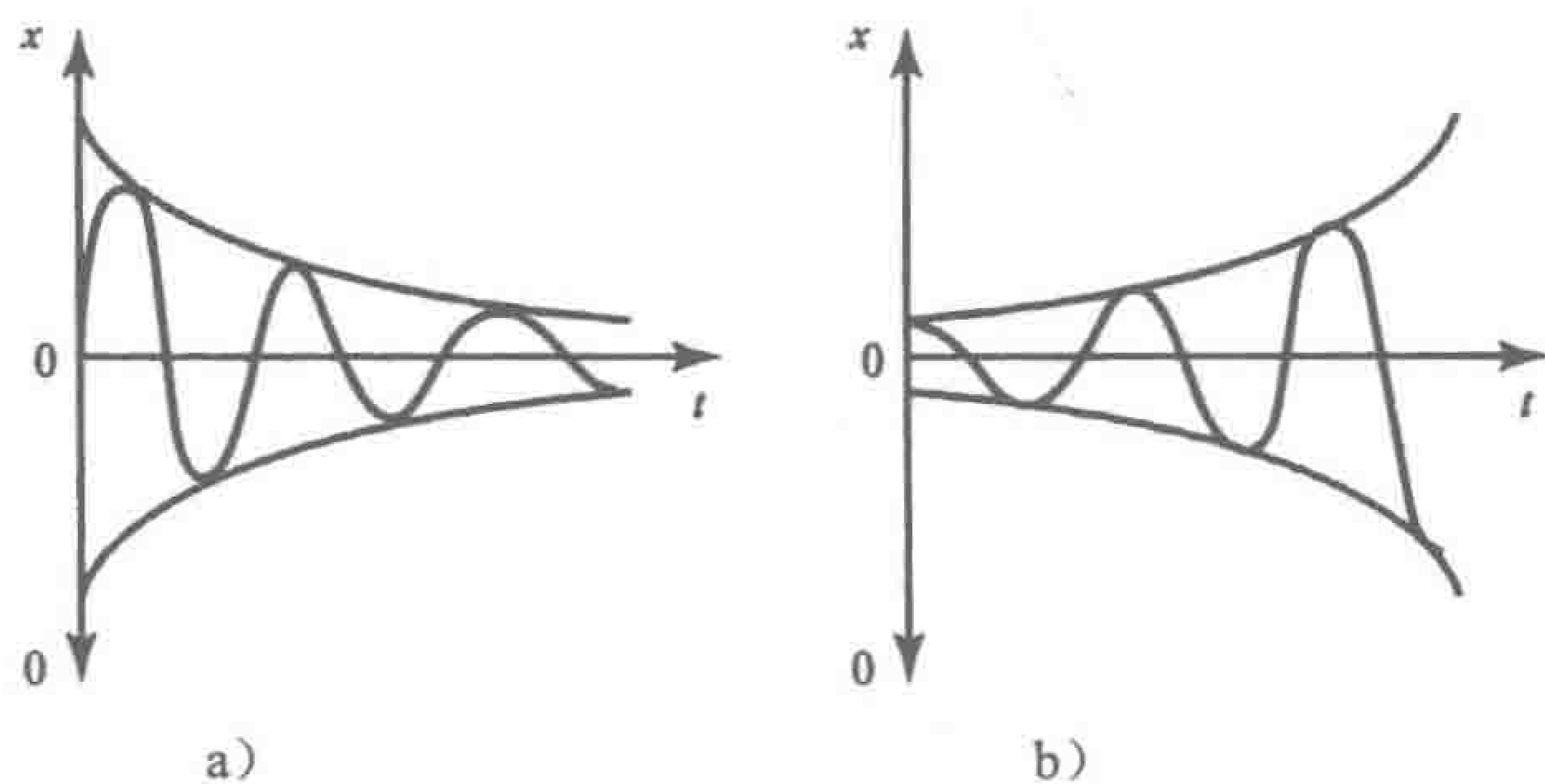


图 20-8 二阶系统

然而，假设系统的传递函数是

$$G(s)=\frac{1}{[s-(2+j1)][s-(2-j1)]}$$

即 $p=+2\pm j1$。当输入是单位脉冲时，输出是 $e^{2t}\sin t$。振荡的振幅是 e^{2t}，随着时间的推移而递增（见图 20-8b），系统是不稳定的。

通常，当脉冲输入到系统时，输出的形式是一系列指数项之和，如果有一项是指数增加的，则输出持续增加，系统是不稳定的。当有一对包含正、负虚部的极点时，输出是振荡的。

如果所有极点的实部是负的，则系统是稳定的。

如果有任何一个极点的实部是正的，则系统是不稳定的。

20.6.1 s 平面

我们可以绘制系统的极点位置图，其 x 轴表示实部，y 轴表示虚部。此图称为 **s 平面**。平面上极点的位置决定了系统的稳定性。图 20-9 说明了 s 平面及根的位置对系统响应的影响。

20.6.2 校正

482 系统的输出可能是不稳定的，或者响应太慢，或者超调太大。系统对输入的响应可以通过加入**补偿器**来改变。补偿器是安装在系统中的模块，它改变了整个系统的传递函数，以此来获得所需的特性。

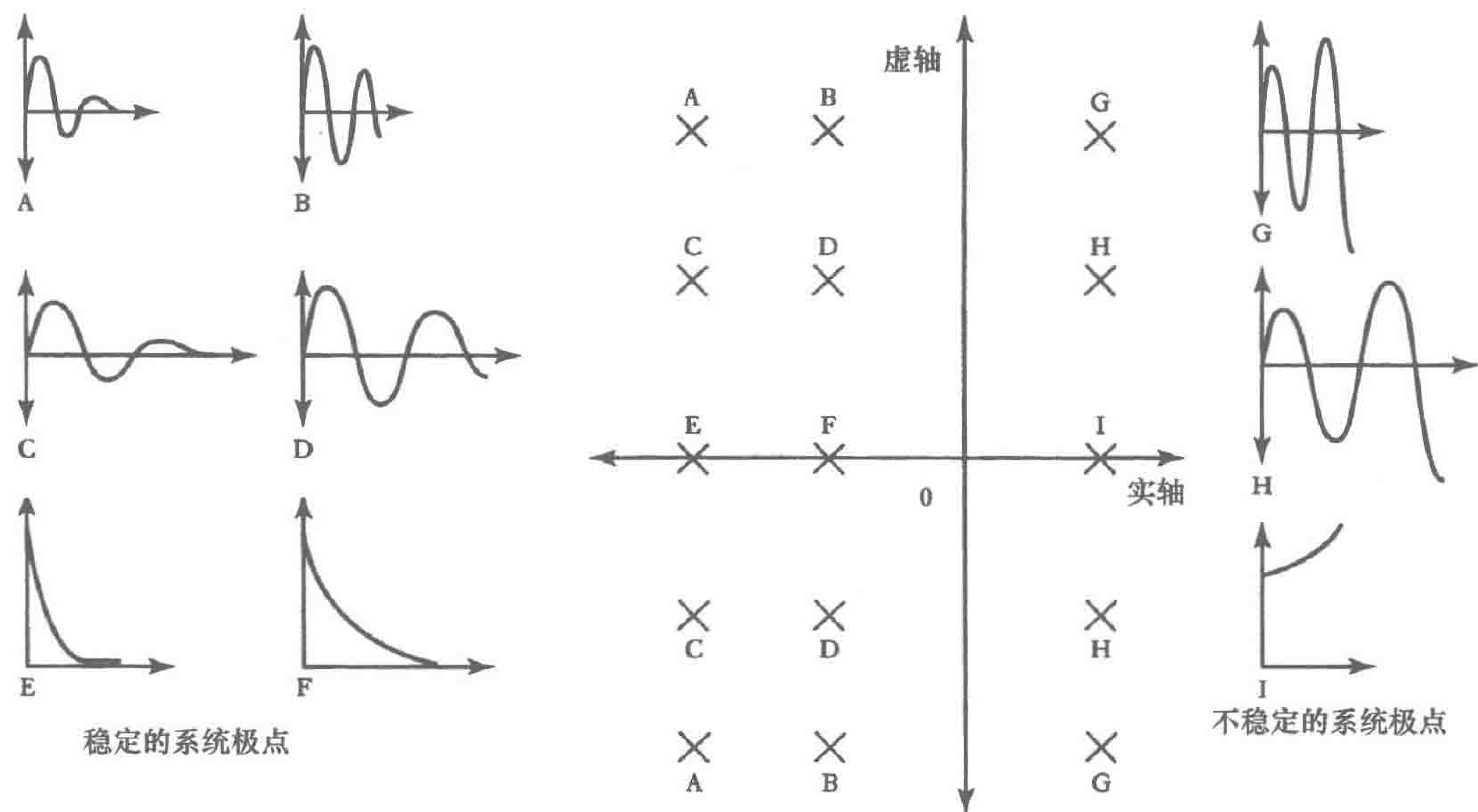


图 20-9　s 平面

作为使用补偿器的例子，考虑具有负反馈通路的位置控制系统，它包括两个前向通道子系统：传递函数是 K 的补偿器和传递函数是 $1/s(s+1)$ 的电动机驱动系统。当系统是临界阻尼时，K 为何值？前向通道的传递函数是 $K/s(s+1)$，反馈通道的传递函数是 1。因此系统的整体传递函数是

$$T(s) = \frac{G(s)}{1 + G(s)H(s)} = \frac{\dfrac{K}{s(s+1)}}{1 + \dfrac{K}{s(s+1)}} = \frac{K}{s(s+1) + K}$$

分母是 $s^2 + s + K$。它的根是

$$s = \frac{-1 \pm \sqrt{1 - 4K}}{2}$$

为了使系统是临界阻尼的，必须有 $1 - 4K = 0$，因此补偿器的比例增益 $K = 1/4$。　483

本章小结

系统的**传递函数** $G(s)$ 是输出的拉普拉斯变换除以输入的拉普拉斯变换。当所有的初始条件是 0 时，为求出微分或积分方程的拉普拉斯变换，可以用 $F(s)$ 代替时间函数 $f(t)$，用 $sF(s)$ 代替函数的一阶导数 $\mathrm{d}f(t)/\mathrm{d}t$，用 $s^2F(s)$ 代替 $\mathrm{d}^2f(t)/\mathrm{d}t^2$，用 $F(s)/s$ 代替 $\int f(t)$ 。

一阶系统的传递函数具有 $G/(\tau s + 1)$ 的形式，其中，τ 是时间常量。**二阶系统**有如下形式的传递函数

$$G(s) = \frac{b_0\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

其中，ζ 是阻尼因子，ω_n 是系统的自然频率。

对于一个传递函数，使其为无穷大的 s 值称为**极点**。它们是特征方程的根。对于一阶系统，如果极点的实部是负数，则系统是稳定的；如果其任何极点的实部是正数，则系统是不稳定的。

习题

20.1　对于以下的输入和输出关系，系统的传递函数是什么？

(a) 输入是 q，输出是 h 的液压系统

$$q = A\frac{\mathrm{d}h}{\mathrm{d}t} + \frac{\rho g h}{R}$$

(b) 输入是 F，输出是 x 的弹簧 - 阻尼 - 质量系统

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + c\frac{\mathrm{d}x}{\mathrm{d}t} + kx = F$$

(c) 输入是 v，输出是 v_C 的 RLC 电路

$$v = RC\frac{\mathrm{d}v_C}{\mathrm{d}t} + LC\frac{\mathrm{d}^2v_C}{\mathrm{d}t^2} + v_C$$

20.2 给定以下的传递函数，系统的时间常量是多少？

(a) $G(s)=5/(3s+1)$　　(b) $G(s)=2/(2s+3)$

20.3 当 $t=0$ 时，有单位阶跃输入，请确定以下系统的输出是如何随时间变化的。

(a) $G(s)=2/(s+2)$　　(b) $G(s)=10/(s+5)$

20.4 对于有以下传递函数的系统，其阻尼状态是什么？

(a) $G(s) = \dfrac{5}{s^2-6s+16}$　　(b) $G(s) = \dfrac{10}{s^2+s+100}$

484

(c) $G(s) = \dfrac{2s+1}{s^2+2s+1}$　　(d) $G(s) = \dfrac{3s+20}{s^2+2s+20}$

20.5 当 $t=0$ 时，有单位阶跃输入，请确定传递函数是 $s/(s+3)^2$ 的系统的输出是什么。

20.6 有单位脉冲输入时，传递函数是 $G=2/[(s+3)\times(s+4)]$的系统的输出是什么？

20.7 对于以下的负反馈系统，系统传递函数是什么？

	前向通道	反馈通道
(a)	$G(s) = \dfrac{4}{s(s+1)}$	$H(s) = \dfrac{1}{s}$
(b)	$G(s) = \dfrac{2}{s+1}$	$H(s) = \dfrac{1}{s+2}$
(c)	$G(s) = \dfrac{4}{(s+2)(s+3)}$	$H(s) = 5$
(d)	两个串联元件 $G_1(s) = 2/(s+2)$ 和 $G_2(s) = 1/s$	$H(s) = 10$

20.8 闭环系统中，前向通道的传递函数是 $5/(s+3)$，负反馈通道的传递函数是 10，则系统的整体传递函数是什么？

20.9 闭环系统的前向通道由两个串联元件组成，传递函数是 5 和 $1/(s+1)$。如果负反馈通道的传递函数是 $2/s$，则系统的整体传递函数是什么？

20.10 闭环系统的前向通道由两个串联元件组成，传递函数是 2 和 $1/(s+1)$。如果负反馈通道传递函数是 s，则系统的整体传递函数是什么？

20.11 系统的传递函数是 $1/[(s+1)(s+2)]$，它的极点是什么？

20.12 以下系统哪个是稳定的，哪个是不稳定的？

(a) $G(s) = 1/[(s+5)(s+2)]$　　(b) $G(s) = 1/[(s-5)(s+2)]$

(c) $G(s) = 1/[(s-5)(s-5)]$　　(d) $G(s) = 1/(s^2+s+1)$

485

(e) $G(s) = 1/(s^2-2s+3)$

第21章　频率响应

本章目的

通过本章的学习，读者将能够：

- 理解频率响应函数的含义。
- 考虑正弦输入下的系统频率响应。
- 伯德图绘制与释义。
- 使用伯德图完成系统辨识。
- 解释术语“带宽”。
- 理解如何通过使用增益裕度和相位裕度来描述系统的稳定性。

21.1　正弦输入

在前面两章中，我们已经考虑过系统对阶跃、脉冲、斜坡输入的响应，本章考虑正弦输入。对于大多数控制系统，正弦输入并不常见。但它是一个有效的输入测试信号，因为系统对于此类输入的响应对评价系统的设计具有非常重要的参考意义。许多常见的信号都可以看成是由一系列的正弦信号叠加得来的。1822 年傅里叶（Jean Baptiste Fourier）提出任何周期性波形，例如方波，都是由正弦信号组合的。通过考虑系统对于各个单独正弦信号的响应，就可以得出系统对更加复杂波形的响应。

21.1.1　系统对正弦输入的响应

考虑一个由微分方程描述的一阶系统：

$$a_1\frac{\mathrm{d}x}{\mathrm{d}t}+a_0x=b_0y$$

其中，y 是输入，x 是输出。假设现有一个单位振幅正弦输入 $y=\sin\omega t$。输出将是什么？当加上 $a_1\mathrm{d}x/\mathrm{d}t$ 和 a_0x 时，就等于正弦信号 $b_0\sin\omega t$。但是正弦信号有一个特性：当求导时，结果仍然是一个正弦信号，且频率不变（余弦信号也是一个正弦信号，即 $\sin(\omega t+90°)$）。不管求导多少次，这条特性一直适用。显然系统的稳态响应 x 也是正弦信号，频率不变，尽管输出可能与输入有不同的振幅和相位。

486

21.2　相量

使用**相量**来描述正弦信号是非常方便的。考虑一个由 $v=V\sin(\omega t+\phi)$ 描述的正弦信号，其中，V 是振幅，ω 是角速度，ϕ 是相角。相量可以用一条长为 $|V|$ 的线段来表示，该线段与相位参考轴的初始角为 ϕ（如图 21-1 所示）。线 $|\ |$ 是用来说明当指定长度时只考虑该振幅或该量的大小。为了完全确定一个相量，需要声明一个振幅和一个角度。一般把相量写成粗斜体，如 $\boldsymbol{V}$。当遇到这样一个符号时，说明这个量具有一个振幅和一个角度。

相量可以用复数来描述。复数可以用 $x+\mathrm{j}y$ 来表示，其中 x 是实部，y 是虚部。在一个虚部为 y 轴、实部为 x 轴的图中，x 和 y 是代表复数的点的笛卡儿坐标（见图 21-2）。

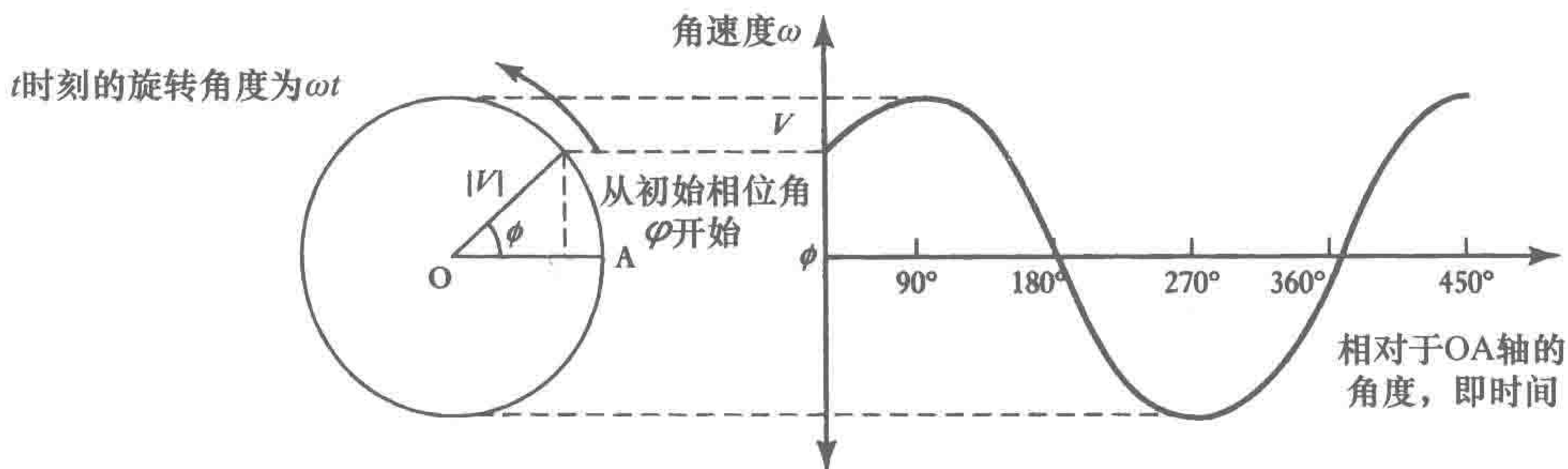


图21-1　正弦信号的相量表示

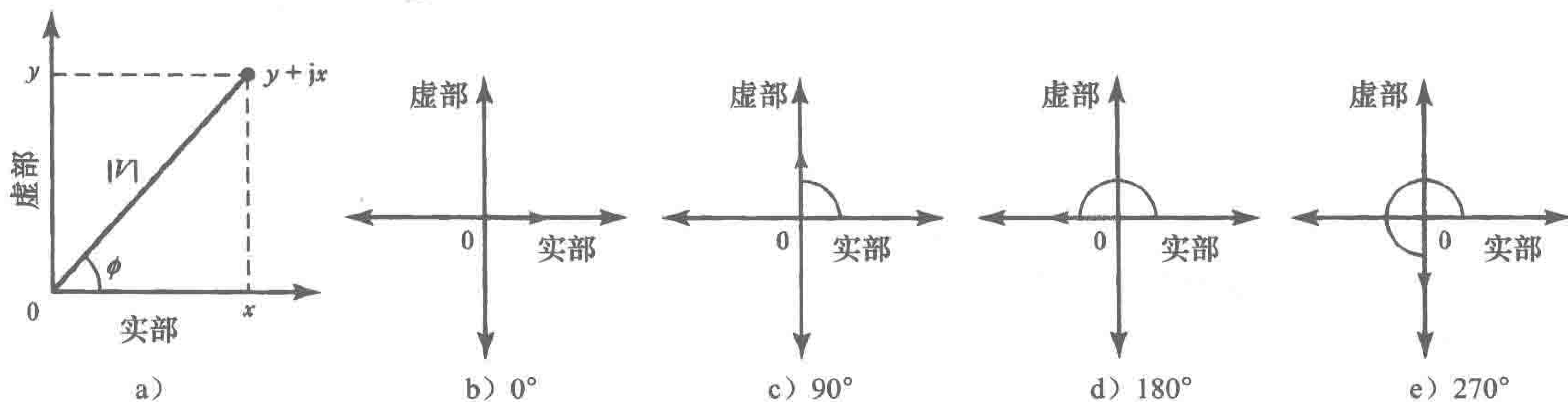


图21-2　相量的复数表示

假如用经过该点和圆心的直线来表示相量，则相量的相角就可以表示为

$$\tan\phi = \frac{y}{x}$$

通过运用勾股定理，长度表示为

487

$$\text{相量的长度}\ |V| = \sqrt{x^2 + y^2}$$

由于 $x = |V|\cos\phi$ 和 $y = |V|\sin\phi$，所以有：

$$V = x + \mathrm{j}y = |V|\cos\phi + \mathrm{j}|V|\sin\phi = |V|(\cos\phi + \mathrm{j}\sin\phi)$$

因此通过对复数实部和虚部的描述来表达一个相量。

考虑一个长为1、相角为0°的相量（见图21-2）。它的复数表示为 $1+\mathrm{j}0$。现在考虑长度相同的相量，但是相角为90°（见图21-2c）。它的复数表示就是 $0+\mathrm{j}1$。这样相量逆时针旋转90°，对应相量乘法就是相量乘以j。如果再把相量旋转90°（见图21-2d），遵循同样的乘法法则，就是原相量乘以 j^2。相量是原相量的反方向，即乘以 -1。$\mathrm{j}^2=-1$，因此 $\mathrm{j}=\sqrt{-1}$。把初始相量旋转270°，即 $3\times90°$，相当于把原相量乘以 $\mathrm{j}^3=\mathrm{j}(\mathrm{j}^2)=-\mathrm{j}$。

为了说明以上内容，考虑一个电压 v，它按下式随时间正弦变化：

$$v = 10\sin(\omega t + 30°)\,\mathrm{V}$$

当表示为相量时，(a) 它的长度是多少？(b) 相对于参考轴的角度是多少？(c) 用复数表示时，实部和虚部是什么？

(a) 相量的长度代表正弦信号的振幅，是10V。

(b) 相量相对于参考轴的角度就是相角，是30°。

(c) 方程给出的实部 $x=10\cos30°=8.7\mathrm{V}$，虚部 $y=10\sin30°=5.0\ \mathrm{V}$。相量可以表示为 $8.7+\mathrm{j}5.0\mathrm{V}$。

21.2.1　相量方程

考虑一个表示单位振幅正弦信号 $x=\sin\omega t$ 的相量。对正弦信号求导有 $\mathrm{d}x/\mathrm{d}t=\omega\cos\omega t$。同样

可以把它写成 $dx/dt=\omega\sin(\omega t+90°)$。换言之，求导只是使相量长度乘以 ω，而角度绕原相量旋转 90°。用复数形式表达，相当于原相量乘以 $j\omega$，因为乘以 j 等价于旋转 90°。 488

因此微分方程

$$a_1\frac{dx}{dt}+a_0x=b_0y$$

可以写成复数形式，成为一个**相量方程**

$$j\omega a_1\boldsymbol{X}+a_0\boldsymbol{X}=b_0\boldsymbol{Y}$$

其中，粗斜体的字母表示这个量是向量。可以说该微分方程已从时域转变为频域。频域的方程可以重新写成

$$(j\omega a_1+a_0)\boldsymbol{X}=b_0\boldsymbol{Y}$$

$$\frac{\boldsymbol{X}}{\boldsymbol{Y}}=\frac{b_0}{j\omega a_1+a_0}$$

但在 20.2 节中，同样的微分方程在 s 域可写成

$$G(s)=\frac{X(s)}{Y(s)}=\frac{b_0}{a_1s+a_0}$$

如果用 $j\omega$ 替代公式中的 s，就得到相同的方程。我们可以用这种方法实现 s 域到频域的转换。这样导出了**频率响应函数**或者**频率传递函数** $G(j\omega)$ 的定义，稳态时有：

$$G(j\omega)=\frac{\text{输出相量}}{\text{输入相量}}$$

为了说明上述内容，考虑如下传递函数的系统频率响应函数：

$$G(s)=\frac{1}{s+1}$$

通过把 s 替换为 $j\omega$，求得其频率响应函数，故

$$G(j\omega)=\frac{1}{j\omega+1}$$

21.3 频率响应

确定系统频率响应的过程如下：

1）把传递函数中的 s 用 $j\omega$ 替换，求得频率响应函数。

2）输出和输入的振幅比是复频响应函数的振幅，即 $\sqrt{x^2+y^2}$。

3）输出和输入之间的相角通过 $\tan\phi=y/x$ 求得，或者通过计算表示频率响应函数的复数的虚部和实部之比得到。

21.3.1 一阶系统的频率响应

一阶系统有如下的传递函数

$$G(s)=\frac{1}{1+\tau s}$$

其中，τ 是系统的时间常量（见 20.2 节）。频率响应函数可以通过把 s 替换成 $j\omega$ 求得，即

489

$$G(j\omega)=\frac{1}{1+j\omega\tau}$$

通过分子、分母同乘以（$1-j\omega\tau$）来简化上式

$$G(j\omega)=\frac{1}{1+j\omega\tau}\times\frac{1-j\omega\tau}{1-j\omega\tau}=\frac{1-j\omega\tau}{1-j^2\omega^2\tau^2}$$

但是 $j^2=-1$，则

$$G(j\omega)=\frac{1}{1+\omega^2\tau^2}-j\frac{\omega\tau}{1+\omega^2\tau^2}$$

这是 $x+jy$ 的格式，因为 $G(j\omega)$ 是输出相量除以输入相量，所以可以求得输出相量除以输入相量的模，写成

$$|G(j\omega)| = \sqrt{x^2+y^2} = \sqrt{\left(\frac{1}{1+\omega^2\tau^2}\right)^2+\left(\frac{\omega\tau}{1+\omega^2\tau^2}\right)^2} = \frac{1}{\sqrt{1+\omega^2\tau^2}}$$

$|G(j\omega)|$ 告诉我们输出相量比输入相量大多少。它通常称为**幅值**或**增益**。输出相量和输入相量间的相位差 ϕ 可以表示成

$$\tan\phi = \frac{y}{x} = -\omega\tau$$

负号表示输出相量滞后于输入相量的角度。

使用上述理论考虑下面的例子。

1）确定有如下传递函数的系统的频率响应函数、增益和相角（电路中有一个电阻串联一个电容，电容两端为输出）。

$$G(s) = \frac{1}{RCs+1}$$

替换 s 为 $j\omega$，得到频率响应函数

$$G(j\omega) = \frac{1}{j\omega RC+1}$$

等式的分子和分母同时乘以 $1-j\omega RC$，重新整理式子得到

$$G(j\omega) = \frac{1}{1+\omega^2(RC)^2} - j\frac{\omega(RC)}{1+\omega^2(RC)^2}$$

故

$$|G(j\omega)| = \frac{1}{\sqrt{1+\omega^2(RC)^2}}$$

且

490

$$\tan\phi = -\omega RC$$

2）当输入为 $2\sin(3t+60°)$ 时，系统的传递函数为

$$G(s) = \frac{4}{s+1}$$

确定系统的输出增益和相角。把 s 替换为 $j\omega$，求得频率响应函数，即

$$G(j\omega) = \frac{4}{j\omega+1}$$

等式的分子和分母同时乘以（$-j\omega+1$），得到

$$G(j\omega) = \frac{-j4\omega+4}{\omega^2+1} = \frac{4}{\omega^2+1} - j\frac{4\omega}{\omega^2+1}$$

增益表示为

$$|G(j\omega)| = \sqrt{x^2+y^2} = \sqrt{\frac{4^2}{(\omega^2+1)^2}+\frac{4^2\omega^2}{(\omega^2+1)^2}} = \frac{4}{\sqrt{\omega^2+1}}$$

相角由 $\tan\phi = y/x$ 给出，故

$$\tan\phi = -\omega$$

对于该输入，有 $\omega = 3\text{rad/s}$。则增益是

$$|G(j\omega)| = \frac{4}{\sqrt{3^2+1}} = 1.3$$

相角由 $\tan\phi = -3$ 求得，即 $\phi = -72°$。这是输入和输出之间的相角。因此输出是 $2.6\sin(3t-72°)$。

21.3.2 二阶系统的频率响应

考虑一个有如下传递函数的二阶系统（见20.3节）：

$$G(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

其中，ω_n 是自然角频率，ζ 是阻尼系数。用 s 替换 $j\omega$ 求得频率响应函数，即

$$G(j\omega)=\frac{\omega_n^2}{-\omega^2+j2\zeta\omega\omega_n+\omega_n^2}=\frac{\omega_n^2}{(\omega_n^2-\omega^2)+j2\zeta\omega_n\omega}$$

$$=\frac{1}{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]+j2\zeta\left(\frac{\omega}{\omega_n}\right)}$$

分子和分母同时乘以

$$\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]-j2\zeta\left(\frac{\omega}{\omega_n}\right)$$

491

得

$$G(j\omega)=\frac{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]-j2\zeta\left(\frac{\omega}{\omega_n}\right)}{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]^2+\left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}$$

这是 $x+jy$ 的格式，因为 $G(j\omega)$ 是输出相量除以输入相量，根据系数 $\sqrt{(x^2+y^2)}$，可得输出相量的大小或幅值大于输入相量的大小或幅值，即

$$|G(j\omega)|=\frac{1}{\sqrt{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]^2+\left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}}$$

输出相量和输入相量的相位差由 $\tan\phi=y/x$ 给出，故：

$$\tan\phi=-\frac{2\zeta\left(\frac{\omega}{\omega_n}\right)}{1-\left(\frac{\omega}{\omega_n}\right)^2}$$

负号表示输出相量滞后于输入相量。

21.4 伯德图

系统的频率响应是当正弦输入信号在一定频率范围内变化时，得到的一系列的增益 $|G(j\omega)|$ 和相角 ϕ 的变化值。这可以表示在两张图中，一张是增益 $|G(j\omega)|$ 随角频率 ω 的变化图，另一张是相角 ϕ 随角频率 ω 的变化图。增益和角频率使用对数刻度。这两幅图称为**伯德图**。

用分贝（dB）表示的增益是

$$\text{用 dB 表示的}\ |G(j\omega)|=20\lg|G(j\omega)|$$

这样，当增益是20dB时

$$20=20\lg|G(j\omega)|$$

$1=\lg|G(j\omega)|$ 和 $10^1=|G(j\omega)|$。这样20分贝的增益就等价于增益为10，因此输出的增益是输入的10倍。增益是40分贝等价于增益为100，即输出的增益是输入的100倍。

21.4.1 $G(s)=K$ 的伯德图

考虑传递函数是 $G(s)=K$ 的系统的伯德图，其中 K 是常量。频率响应函数是 $G(j\omega)=K$。增益 $|G(j\omega)|=K$，因此用分贝表示，$|G(j\omega)|=20\lg K$。增益曲线是一条常量线，改变 K 仅仅使增益线上下移动一定的分贝数。相角是0。图21-3显示了该伯德图。

492

21.4.2　$G(s)=1/s$ 的伯德图

考虑传递函数是 $G(s)=1/s$ 的系统的伯德图。频率响应函数 $G(j\omega)$ 是 $1/j\omega$。分子分母同时乘以 j，有 $G(j\omega)=-j/\omega$。增益 $|G(j\omega)|=1/\omega$。用分贝表示就是 $20\lg(1/\omega)=-20\lg\omega$。当 $\omega=1$ rad/s 时，增益是 0dB。当 $\omega=10$ rad/s 时，增益是 -20dB。当 $\omega=100$ rad/s 时，增益是 -40dB。角频率每增加 10 倍，增益降低 -20dB。增益曲线是一条斜率为 -20dB 每 10 倍频程（或 dB/decade）的直线，它在 $\omega=1$ rad/s 时经过 0dB。相角由下式给出：

$$\tan\phi=\frac{-\frac{1}{\omega}}{0}=-\infty$$

因此，对于所有频率，相角 $\phi=-90°$。图 21-4 显示了该伯德图。

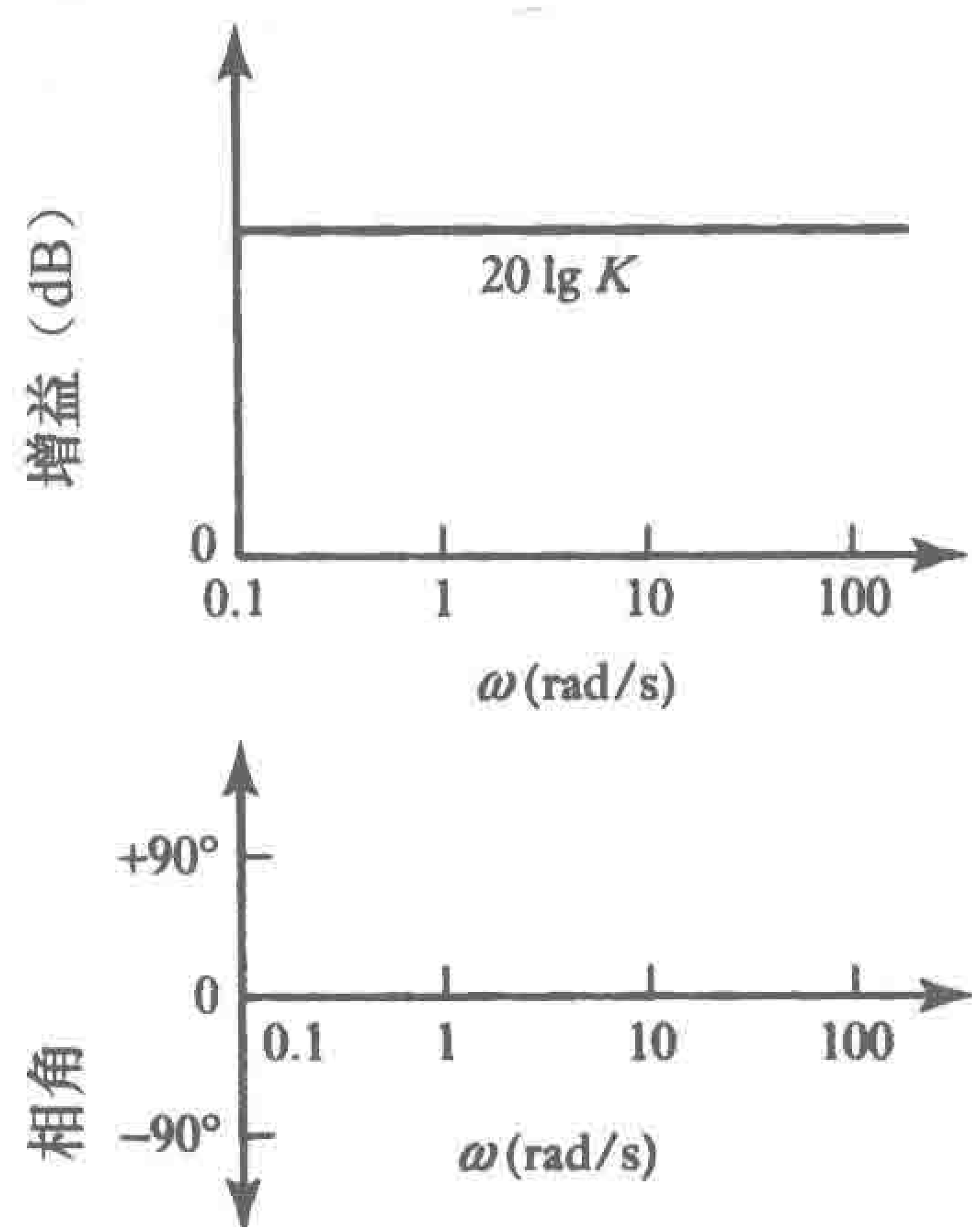


图 21-3　$G(s)=K$ 的伯德图

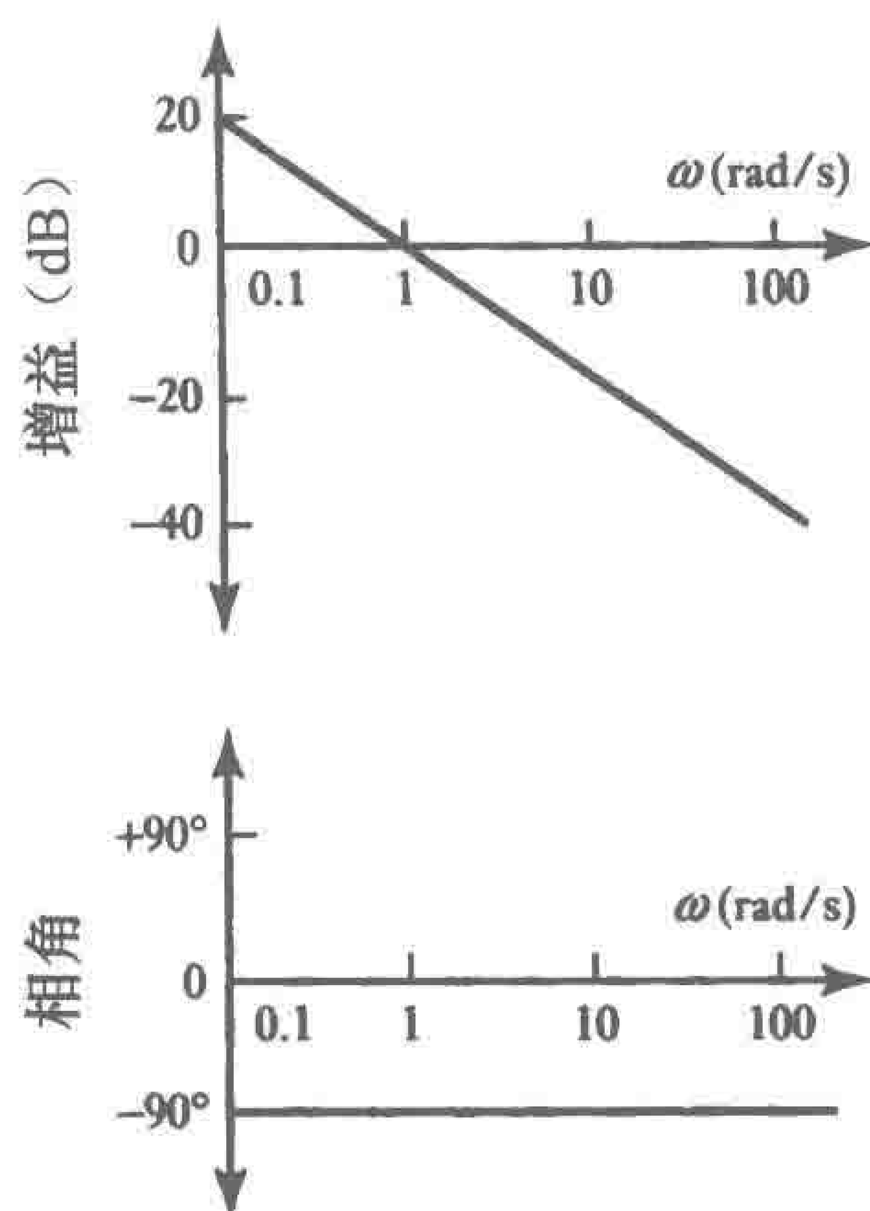


图 21-4　$G(s)=1/s$ 的伯德图

21.4.3　一阶系统的伯德图

考虑传递函数是 $G(s)=1/(\tau s+1)$ 的一阶系统的伯德图。频率响应函数是

$$G(j\omega)=\frac{1}{j\omega\tau+1}$$

增益（见 21.3.1 节）是

$$|G(j\omega)|=\frac{1}{\sqrt{1+\omega^2\tau^2}}$$

用分贝表示，就是

$$20\lg\left(\frac{1}{\sqrt{1+\omega^2\tau^2}}\right)$$

当 $\omega\ll1/\tau$ 时，$\omega^2\tau^2$ 远小于 1，所以增益 20lg1 = 0dB。因此在低频时，增益曲线是 0dB 的直线。在高频时，$\omega\gg1/\tau$，$\omega^2\tau^2$ 远大于 1，分母中的 1 可以忽略不计。增益是 $20\lg(1/\omega\tau)$，即 $-20\lg(\omega\tau)$。增益曲线是一条斜率为 -20dB/decade（分贝/10 倍频程）的直线，它在 $\omega\tau=1$ 时与 0dB 线相交，即 $\omega=1/\tau$。图 21-5显示低频及高频时的这些线及其交点，称为**拐点或转角频率**（$\omega=1/\tau$）。这两条直线称为真实曲线的渐次逼近曲线。真实曲线把两条直线的交点修圆，与逼近曲线的增益差在拐点处最大，为 3dB。

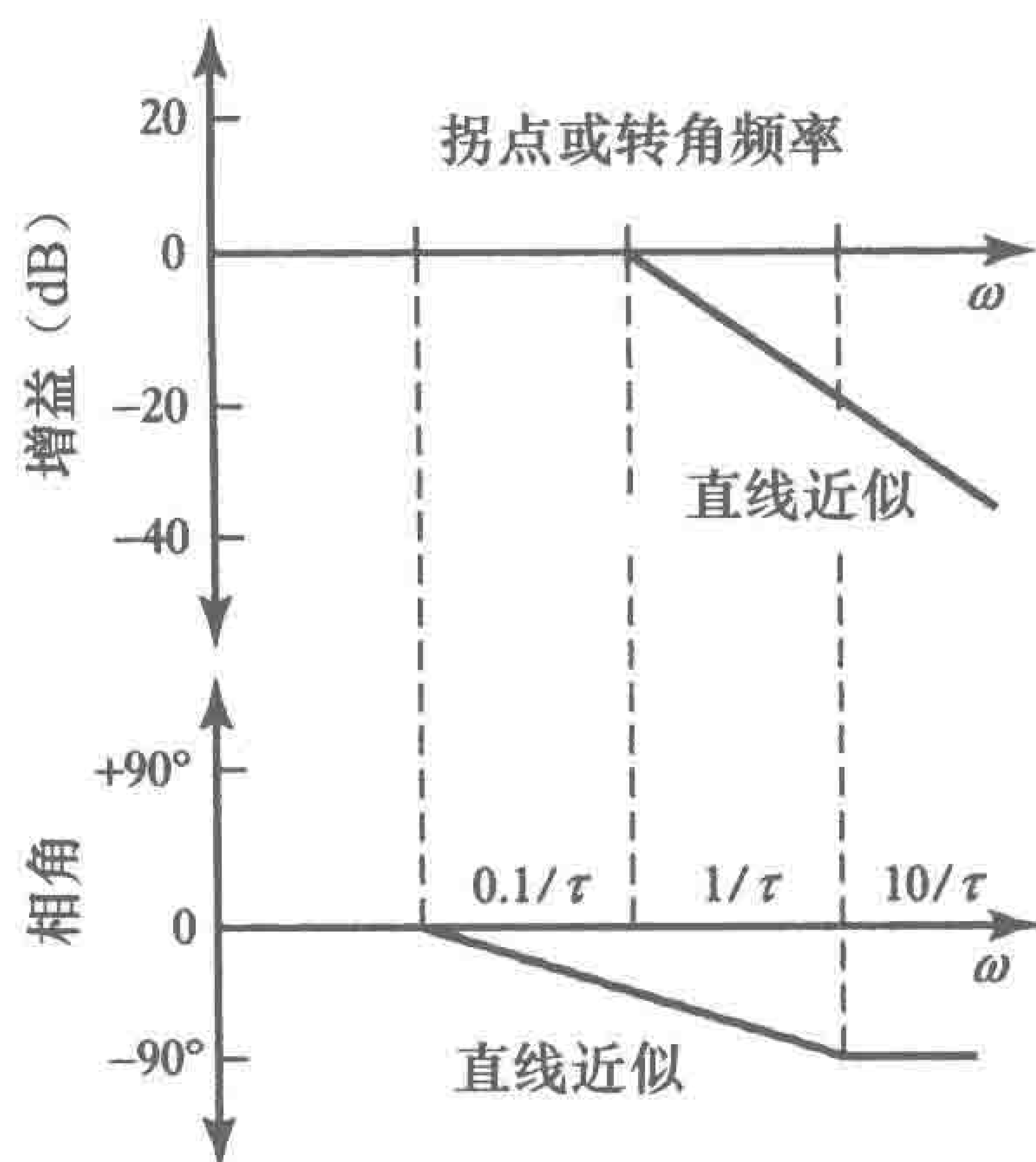


图 21-5　一阶系统的伯德图

一阶系统的相角由 $\tan\phi = -\omega\tau$ 求得（见21.2.1节）。在低频时，当 $\omega < 0.1/\tau$ 时，相角约为0°。在高频时，当 $\omega > 10/\tau$ 时，相角约为 $-90°$。在二者之间，相角可认为是伯德图中的一条直线（见图21-5）。假定直线的最大误差是5.5°。

例子中的系统是一个 RC 滤波器（见20.2.2节），即电阻 R 与电容 C 串联，输出是电容两端的电压。传递函数是 $1/(RCs+1)$，频率响应函数是 $1/(\mathrm{j}\omega\tau+1)$，其中 $\tau = RC$。伯德图如图21-5所示。

21.4.4 二阶系统的伯德图

考虑一个二阶系统，其传递函数是

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

用 $\mathrm{j}\omega$ 替代 s 求得频率响应函数，即

$$G(\mathrm{j}\omega) = \frac{\omega_n^2}{-\omega^2 + \mathrm{j}2\zeta\omega_n\omega + \omega_n^2}$$

增益是（见21.3.2节）

$$|G(\mathrm{j}\omega)| = \frac{1}{\sqrt{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}}$$

493～494

因此用分贝表示的增益是

$$20\lg\frac{1}{\sqrt{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}}$$

$$= -20\lg\sqrt{\left[1-\left(\frac{\omega}{\omega_n}\right)^2\right]^2 + \left[2\zeta\left(\frac{\omega}{\omega_n}\right)\right]^2}$$

当 $(\omega/\omega_n) \ll 1$ 时，增益约为 $-20\lg 1$ 或为0dB。当 $(\omega/\omega_n) \gg 1$ 时，增益约为 $-20\lg(\omega/\omega_n)^2$。因此，当 ω 以10倍增加时，增益增加 $-20\lg 100$ 或者 -40dB。在低频时，增益曲线是0dB的直线；而在高频时，增益是斜率为 -40dB/decade 的直线。两条直线的交点，即拐点，是在 $\omega = \omega_n$ 处。增益曲线由以上两条渐近线所近似。然而，其真实值由阻尼比 ζ 确定。图21-6显示了两条渐近线及在不同阻尼比下的真实曲线。

相角由下式给出（见21.3.2节）。

$$\tan\phi = -\frac{2\zeta\left(\frac{\omega}{\omega_n}\right)}{1-\left(\frac{\omega}{\omega_n}\right)^2}$$

当 $\omega/\omega_n \ll 1$ 时，如 $\omega/\omega_n = 0.2$，则 $\tan\phi$ 约为0，$\phi = 0°$。当 $\omega/\omega_n \gg 1$，如 $\omega/\omega_n = 5$，则 $\tan\phi$ 约为 $-\frac{2\zeta\times 5}{1-25}$，$\phi = -180°$。当 $\omega = \omega_n$ 时，则 $\tan\phi$ 约为 $-\infty$，$\phi = -90°$。因此，一条合理的近似曲线是在 $\omega = \omega_n$ 时通过 $-90°$，在 $\omega/\omega_n = 0.2$ 时通过0°，在 $\omega/\omega_n = 5$ 时通过 $-180°$（如图21-6所示）。

21.4.5 建立伯德图

考虑一个由一系列的元素串联组成的系统。系统的整体传递函数是（见20.4节）：

$$G(s) = G_1(s)G_2(s)G_3(s)\cdots$$

因此，对于一个两元系统的频率响应函数，用 $\mathrm{j}\omega$ 替代 s，有

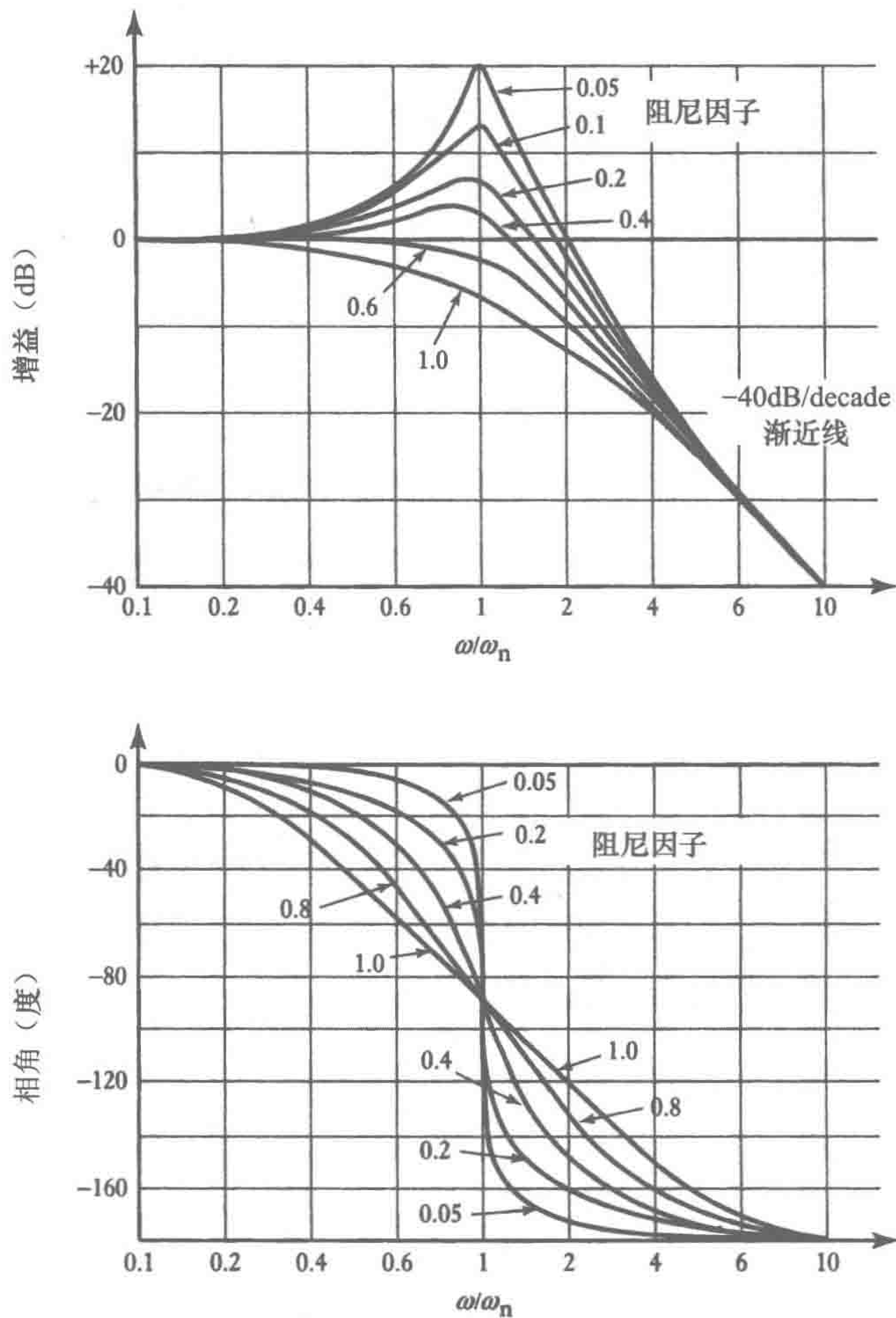


图 21-6 二阶系统的伯德图

$$G(j\omega) = G_1(j\omega)G_2(j\omega)$$

把传递函数 $G_1(j\omega)$ 用复数形式表示（见 21.2 节）：

$$x + jy = |G_1(j\omega)|(\cos\phi_1 + j\sin\phi_1)$$

其中 $|G(j\omega)|$ 是该频率响应函数的增益，ϕ_1 是相角。同样，我们可以把 $G_2(j\omega)$ 写成

$$|G_2(j\omega)|(\cos\phi_2 + j\sin\phi_2)$$

因此

$$G(j\omega) = |G_1(j\omega)|(\cos\phi_1 + j\sin\phi_1) \times |G_2(j\omega)|(\cos\phi_2 + j\sin\phi_2)$$

$$= |G_1(j\omega)||G_2(j\omega)|[\cos\phi_1\cos\phi_2 + j(\sin\phi_1\cos\phi_2 + \cos\phi_1\sin\phi_2) + j^2\sin\phi_1\sin\phi_2]$$

但 $j^2 = -1$，且因为 $\cos\phi_1\cos\phi_2 - \sin\phi_1\sin\phi_2 = \cos(\phi_1 + \phi_2)$ 和 $\sin\phi_1\cos\phi_2 + \cos\phi_1\sin\phi_2 = \sin(\phi_1 + \phi_2)$，所以

$$G(j\omega) = |G_1(j\omega)||G_2(j\omega)|[\cos(\phi_1 + \phi_2) + j\sin(\phi_1 + \phi_2)]$$

系统的频率响应函数的增益是一组独立系统的增益的乘积，相角是一组独立系统的相角之和，即

495
~
496

$$|G(j\omega)| = |G_1(j\omega)||G_2(j\omega)||G_3(j\omega)|\cdots$$

$$\phi = \phi_1 + \phi_2 + \phi_3 + \cdots$$

现在，考虑增益对数曲线的伯德图：

$$\lg|G(j\omega)| = \lg|G_1(j\omega)| + \lg|G_2(j\omega)| + \lg|G_3(j\omega)| + \cdots$$

这样我们可以通过把组成元素的增益曲线的伯德图相加求得系统的伯德图。同样，相角曲线可以通过把组成元素的相角曲线相加求得。

通过使用许多基本元素，系统大范围的伯德图可以很容易求得。如下基本元素被使用了：

1）$G(s)=K$，伯德图如图 21-3 所示。

2）$G(s)=1/s$，伯德图如图 21-4 所示。

3）$G(s)=s$，伯德图是图 21-4 曲线的镜像图，$|G(j\omega)|=20\text{dB/decade}$，当 $\omega=1\text{rad/s}$ 时过 0dB。ϕ 是 90°常量。

4）$G(s)=1/(\tau s+1)$，伯德图如图 21-5 所示。

5）$G(s)=\tau s+1$，伯德图是图 21-5 曲线的镜像图。对于增益曲线，拐点在 $1/\tau$，在此之前直线在 0dB 处。在此之后直线斜率是 20dB/decade。$0.1/\tau$ 时，相角是 0°，在 $10/\tau$ 时升至 +90°。

6）$G(s)=\omega_n^2/(s^2+2\zeta\omega_n s+\omega_n^2)$，伯德图如图 21-6 所示。

7）$G(s)=(s^2+2\zeta\omega_n s+\omega_n^2)/\omega_n^2$，伯德图是图 21-6 的镜像图。

为了说明上述内容，考虑绘制系统的渐近伯德图，它的传递函数是

$$G(s)=\frac{10}{2s+1}$$

传递函数由两个基本元素组成，一个传递函数是 10，另一个传递函数是 $1/(2s+1)$。伯德图可由单独绘制的两张伯德图，然后将它们相加得到需要的伯德图。传递函数是 10 的伯德图如图 21-3 所示，其中 $K=10$。传递函数 $1/(2s+1)$ 的伯德图如图 21-5 所示，其中 $\tau=2$。结果显示在图 21-7 中。

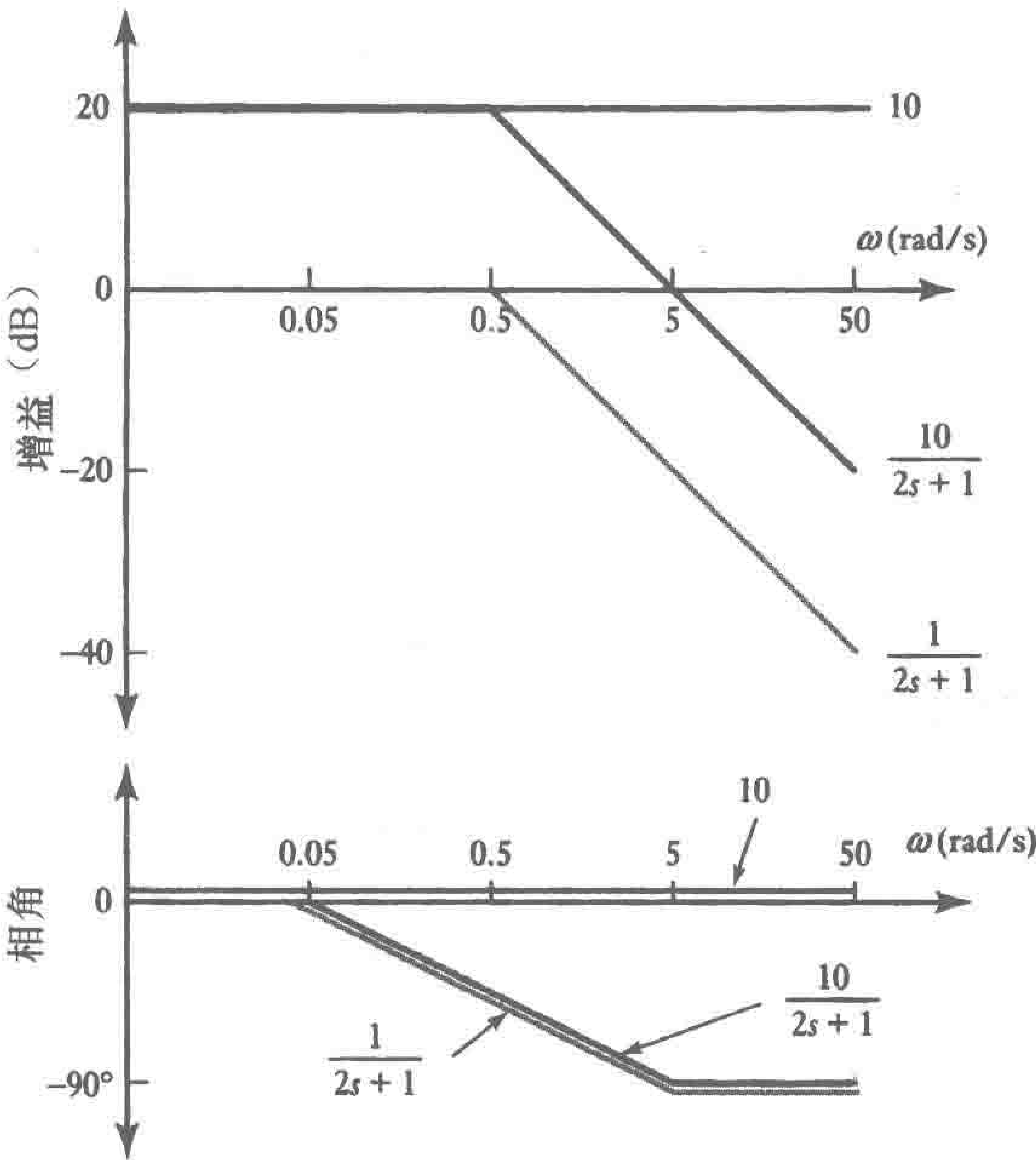


图 21-7 建立伯德图

再举一个例子，考虑绘制另一个系统的渐近伯德图，它的传递函数是

$$G(s)=\frac{2.5}{s(s^2+3s+25)}$$

传递函数由3个基本元素组成：一个传递函数是0.1，一个传递函数是$1/s$，一个传递函数是$25/(s^2+3s+25)$。传递函数0.1的伯德图如图21-3所示，其中$K=0.1$。传递函数$1/s$的伯德图如图21-4所示。传递函数$25/(s^2+3s+25)$可用$\omega_n^2/(s^2+2\zeta\omega_n s+\omega_n^2)$表示，其中$\omega_n=5\text{rad/s}$，$\zeta=0.3$。拐点在$\omega=\omega_n=5\text{rad/s}$。相角曲线的渐近曲线在拐点处过$-90°$，当$\omega/\omega_n=0.2$时，经过$0°$，当$\omega/\omega_n=5$时，经过$-180°$。图21-8是最终的曲线。

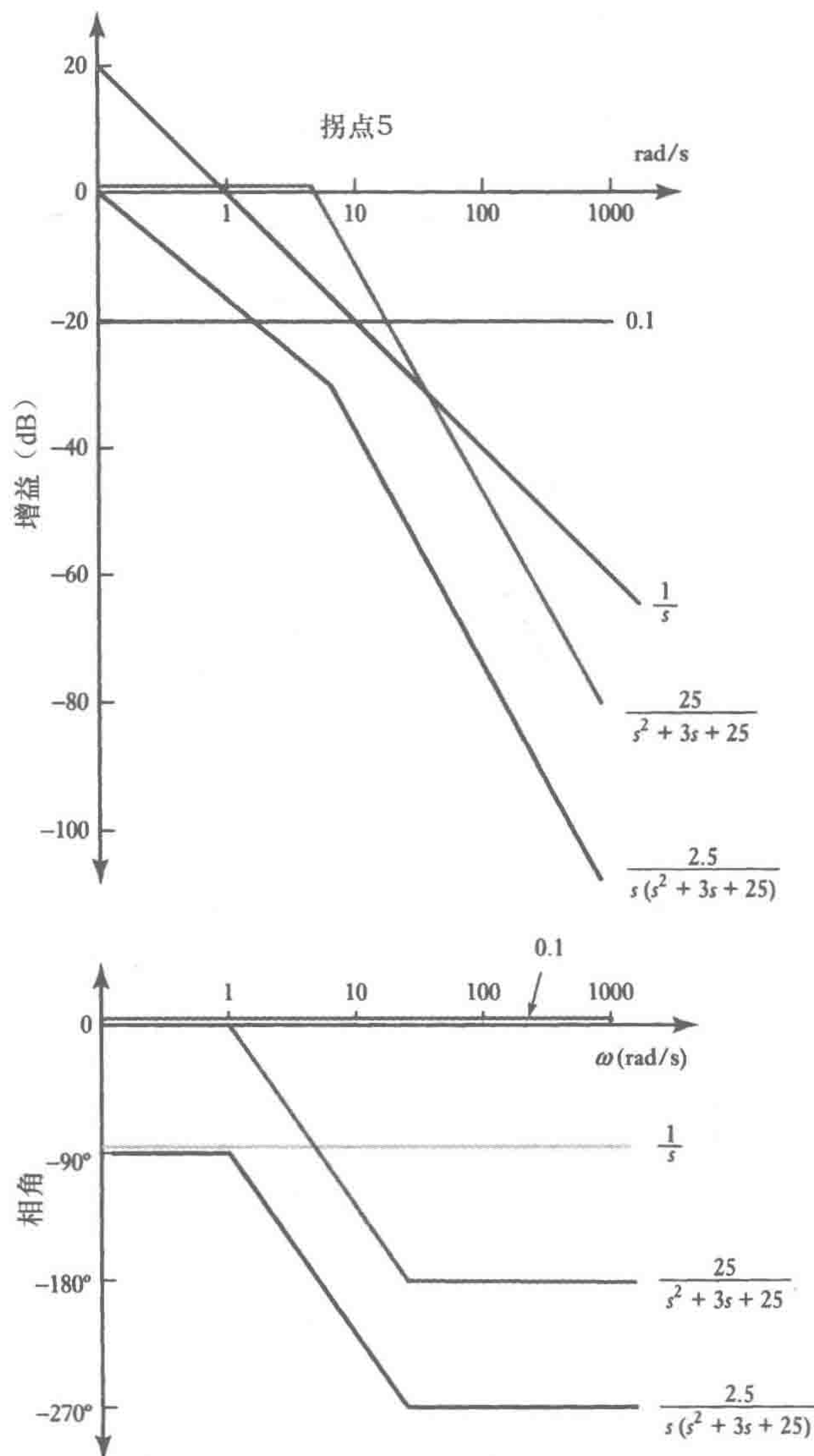


图21-8　建立伯德图

上述通过组成元素求得伯德图的方法，运用直线逼近，曾经广泛运用，但是在现在的计算机时代却是不太必要的。

21.4.6　系统辨识

如果我们能通过分析正弦输入的响应用实验方法确定系统的伯德图，那么我们就可以获得系统的传递函数。基本上，我们可以绘出增益伯德图的渐近线，并得到它的梯度。相角曲线可用于验证增益分析结果的正确性。

1）如果低频的梯度（转角频率之前）是0，那么传递函数中就没有s或者$1/s$元素。传递

函数分子中的 K 元素可以从低频增益算出，分贝表示的增益是 $20\lg K$。

2）如果低频时的初始梯度是 -20dB/decade，那么传递函数有 $1/s$ 元素。

3）如果在转角频率的梯度变得更进一步负 20dB/decade，那么传递函数的分母有 $1+s/\omega_c$ 项，ω_c 是转角频率，在此处发生变化。对于多个转角频率，可发生这种环节。

4）如果在转角频率的梯度变得更进一步正 20dB/decade，传递函数的分子有 $1+s/\omega_c$ 项，ω_c 是转角频率，在此处发生变化。对于多个转角频率，可发生这种环节。

5）如果在转角频率的梯度变得更进一步负 40dB/decade，传递函数的分母有（$s^2/\omega_c^2+2\xi s/\omega_c+1$）项，阻尼比 ζ 可以通过考虑伯德图在转角频率处的细节求得，如图 21-6 所示。

6）如果在转角频率的梯度变得更进一步正 40dB/decade，传递函数的分子有（$s^2/\omega_c^2+2\xi s/\omega_c+1$）项，阻尼比 ζ 可以通过考虑伯德图转角频率处的细节求得，如图 21-6 所示。

7）如果低频梯度不是 0，则传递函数分子中的 K 可以由低频渐近曲线判断求得。在低频时，传递函数的许多项可以忽略，分贝增益约为 $20\lg(K/\omega^2)$。因此在 $\omega=1$ 时分贝增益约为 $20\lg K$。

497 ~ 499

下面举例说明如何使用上述理论，考虑图 21-9 的增益伯德图。初始梯度是 0，故传递函数中没有 $1/s$ 或 s 项。初始的增益是 20dB，所以 $20=20\lg K$ 和 $K=10$。在 $\omega=10\text{rad/s}$ 后，梯度变为 -20dB/decade。故分母中有（$1+s/10$）项，传递函数是 $10/(1+0.1s)$。

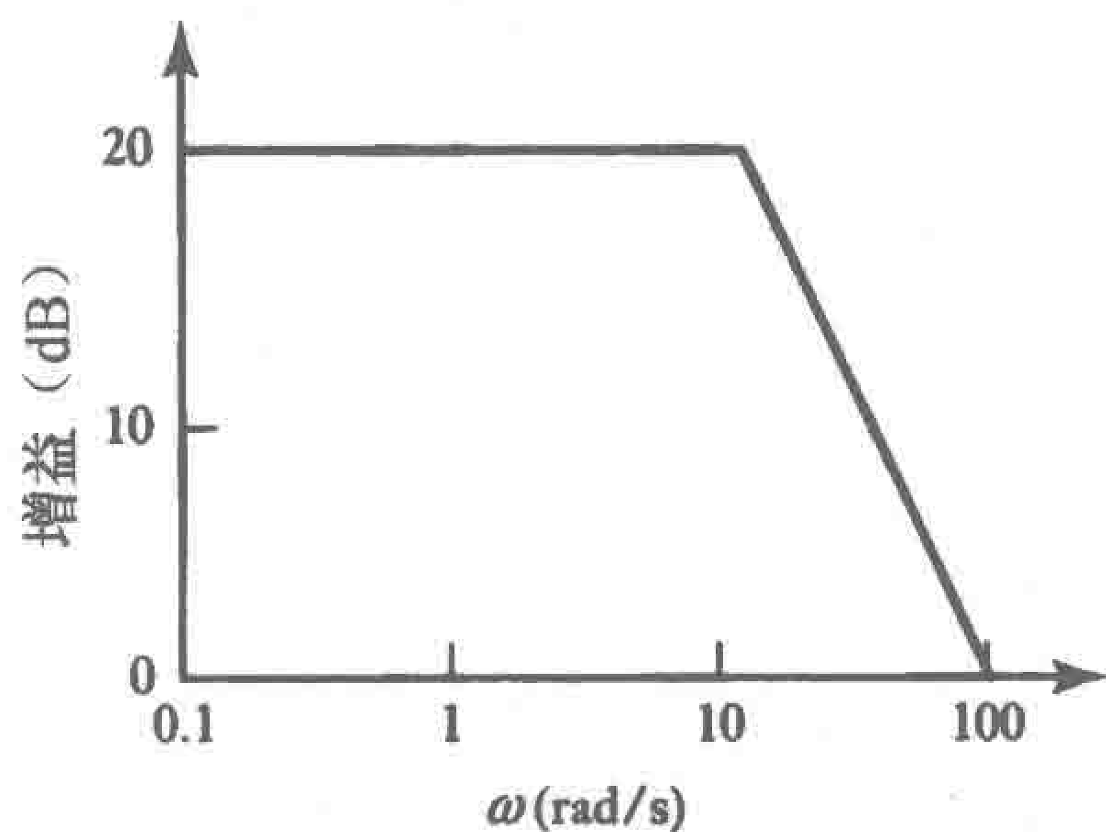


图 21-9 伯德图

再比如说，考虑图 21-10。有个 -20dB/decade 的初始斜坡，故有 $1/s$ 项。在转角频率 1.0rad/s 处，梯度有 -20dB/decade 的变化，故有一个 $1/(1+s/1)$ 项；在转角频率 10rad/s 处，梯度又有 -20dB/decade 的变化，故有一个 $1/(1+s/10)$ 项。当 $\omega=1$ 时，增益是 6dB，故 $6=20\lg K$，$K=10^{6/20}=2.0$。因此传递函数是 $2.0/s(1+s)(1+0.1s)$。

再比如说，图 21-11 的伯德图说明初始梯度为 0，在 10rad/s 时变为 -40dB/decade。初始的增益是 10dB，故 $10=20\lg K$，$K=10^{0.5}=3.2$。在 10rad/s 时变为 -40dB/decade，故系统传递函数的分母有（$s^2/10^2+2\xi s/10+1$）项。因此，传递函数是 $3.2/(0.01s^2+0.2\xi s+1)$。通过与图 21-6 在转角频率处的伯德图比较，求得阻尼因子。转角处上升 6dB，故阻尼因子约为 0.2。因此传递函数是 $3.2/(0.01s^2+0.04s+1)$。

500

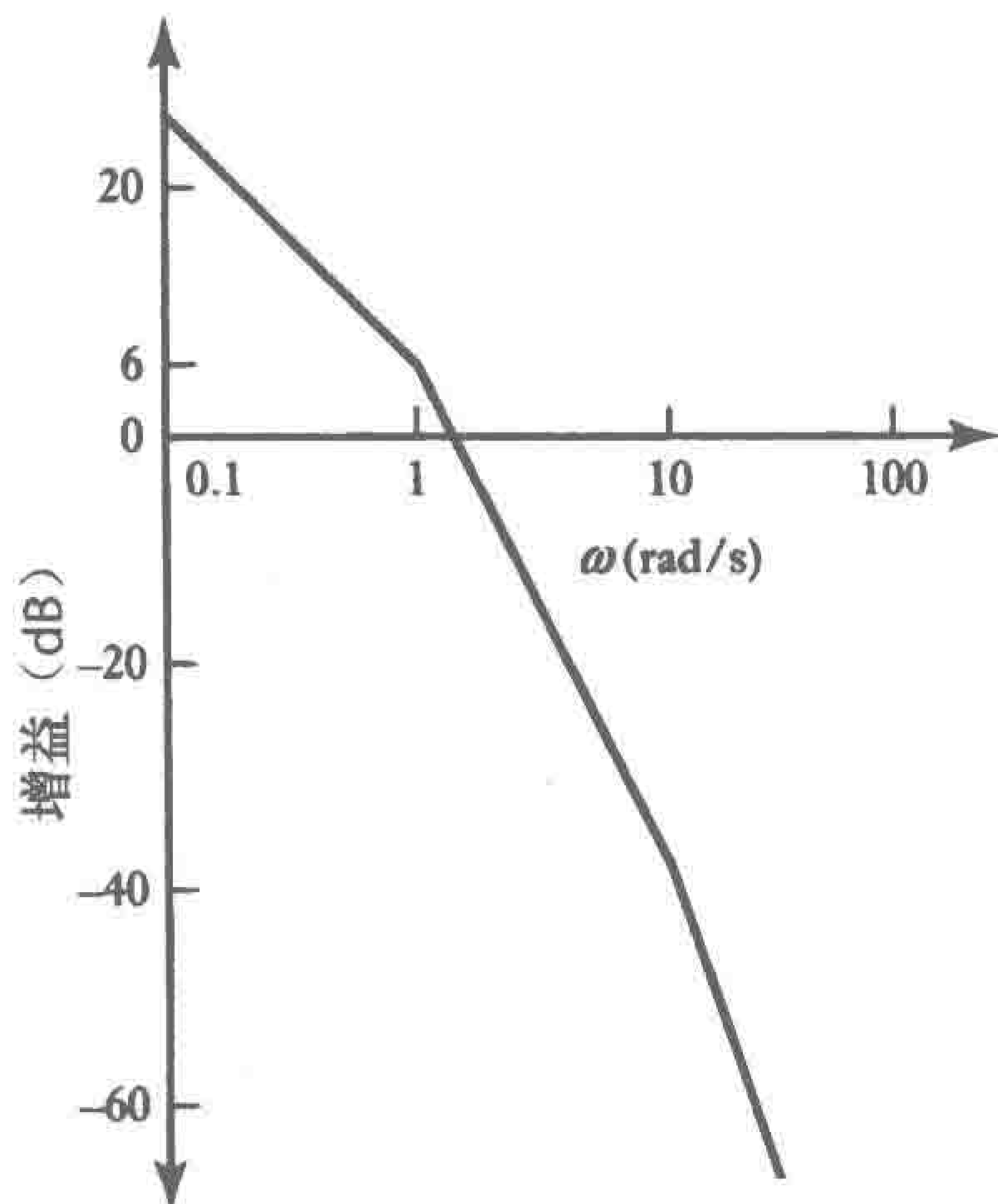


图 21-10 伯德图 1

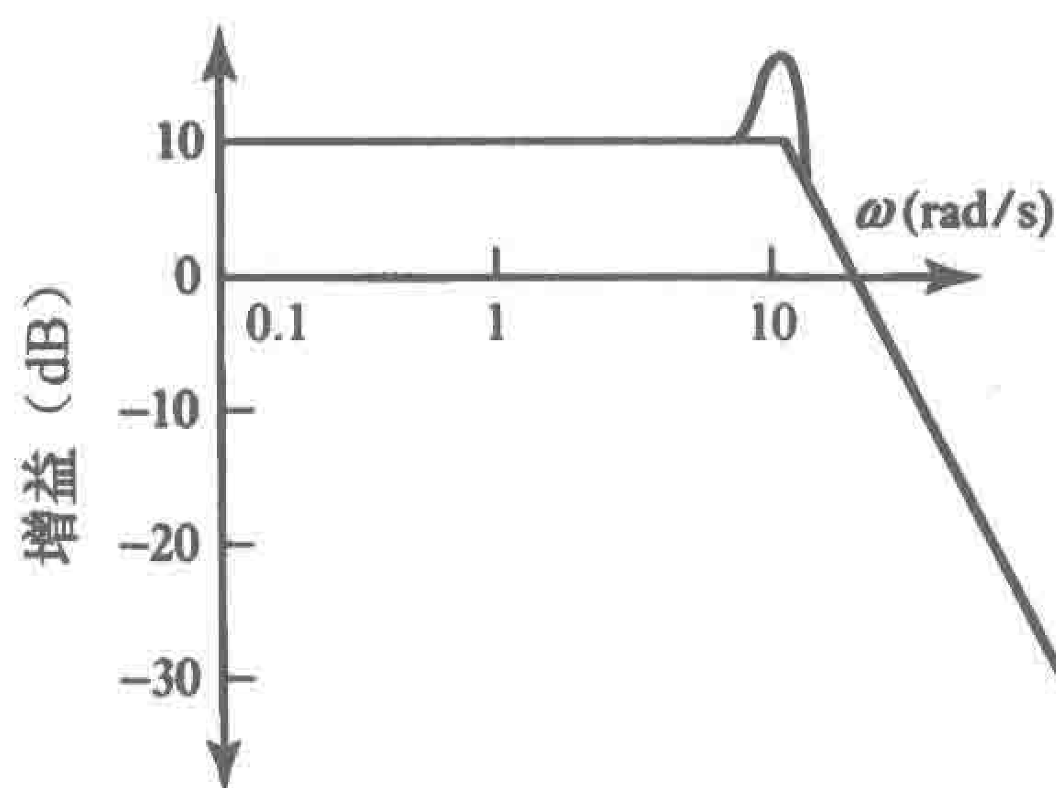


图 21-11 伯德图 2

21.5　性能指标

用来描述系统对于正弦输入的响应特性的参数是谐振峰值和带宽。**谐振峰值** Mp 定义为增益的最大值（见图 21-12）。谐振峰值的值对应于系统的最大超调的大值。对于一个二阶系统来说，它可以直接与比较图 21-6 中伯德图和响应发现它与阻尼系数的关系，一个小阻尼系数对应于高的谐振峰值。

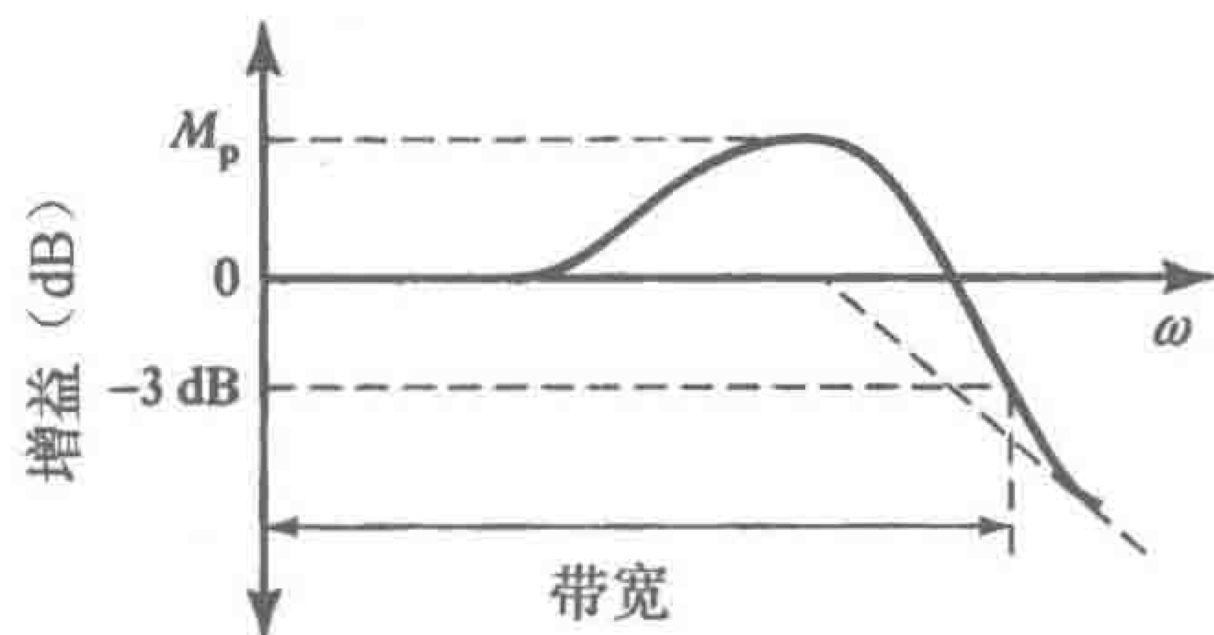


图 21-12　性能指标

带宽定义为增益不小于 −3dB 的频率带。在增益为 −3dB处的频率称为截止频率。增益用分贝表示：

$$|G(j\omega)| = 20\lg_{10}|G(j\omega)|$$

因此

$$-3 = 20\log_{10}|G(j\omega)|$$

有 $|G(j\omega)| = 0.707$，即振幅降到初始幅值的 0.707 倍。由于一个正弦波的功率与它振幅的平方成正比，所以此时功率降为初值的一半（$0.707 \times 0.707 = 0.5$）。因此，在 −3dB 的截止线是指在该处输入信号功率衰减为原来一半。如图 21-12 所示的系统伯德图，带宽是从频率为 0 处到增益降为 −3dB 处的频率。这是一个典型的测量系统，它在低频时常常不表现出衰减，在高频时才表现出增益的降低。

作为说明，20.2.2 节项 1 中描述的例子，系统（一个电路中电阻和电容串联，把电容两端的电压作为输出）的传递函数是 $G(s) = 1/(RCs+1)$，系统增益为：

501

$$|G(j\omega)| = \frac{1}{\sqrt{1+\omega^2(RC)^2}}$$

对于增益比 0.707，截止频率 ω_c 是

$$0.707 = \frac{1}{\sqrt{1+\omega_c^2(RC)^2}}$$

$$1+\omega_c^2(RC)^2 = (1/0.707)^2 = 2$$

因此，$\omega_c = 1/RC$。这样的电路称为低通滤波器，因为低频信号通过该滤波器到输出只有很少的衰减，而较高频率的信号则衰减了。

21.6　稳定性

当正弦信号输入系统时，系统的输出也是正弦信号，其角频率不变，振幅和相角发生改变。考虑图 21-13 中具有负反馈的无输入闭环系统。假设有一个半波整流的正弦脉冲作为系统的误差信号，将它传递到输出并反馈到比较器。此时信号的振幅不变，但是落后了半个周期，即相角改变了 180°（如图 21-13 所示）。当该信号从输入信号中减去时，最终得到的误差信号延续了初始的半波整流脉冲。这种脉冲返回到反馈回路，再次延续了该信号。这样我们就得到了一个自持振荡。

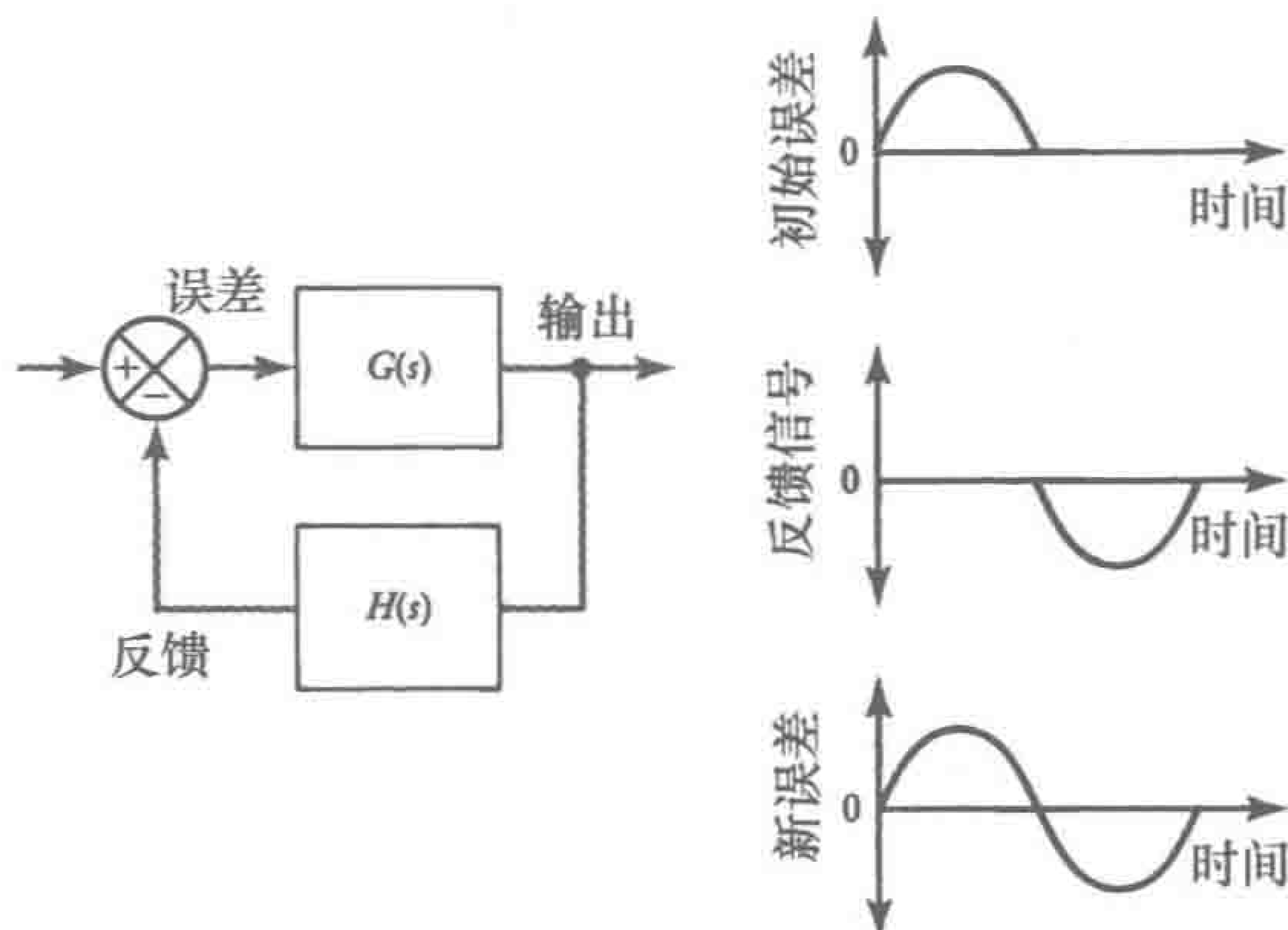


图 21-13　自持振荡

对于自持振荡，必然有一个增益是 1，相角是 −180°的系统频率响应函数。信号传递的

系统是 $G(s)$ 与 $H(s)$ 串联组成的。如果增益小于1，则每个后续的半波脉冲就会变小，故振荡逐渐衰减。如果增益大于1，则每个后续的半波脉冲就会比前一个大，故波形越来越振荡，系统不稳定。

502

1）如果由 $G(s)$ 与 $H(s)$ 串联组成的系统的增益等于1，相角是 $-180°$，则控制系统会在一个不变的振幅下不断振荡。

2）如果由 $G(s)$ 与 $H(s)$ 串联组成的系统的增益小于1，相角是 $-180°$，则控制系统会在一个递减的振幅下振荡。

3）如果由 $G(s)$ 与 $H(s)$ 串联组成的系统的增益大于1，相角是 $-180°$，则控制系统会在一个递增的振幅下不断的振荡，系统是不稳定的。

一个好的稳定的控制系统常常需要 $G(s)H(s)$ 的增益显著小于1。一般常用的取值为0.4～0.5。另外，相角需要为 $-115°$ ～ $-125°$。这些数值可以产生一个轻微欠阻尼的控制系统。该系统在阶跃信号的输入下，会有20%～30%的超调（见19.5节中的相关解释）。

对于一个控制系统，我们关心的是系统的稳定性，并且在一些小干扰下不会产生明显的振荡。**增益裕度**（gain margin）是相角为 $-180°$ 时增益比变为1需要乘以的比例系数，即处于临界不稳定处。**相位裕度**（phase margin）是增益为1时相角与 $-180°$ 相差的角度。一个良好的稳定的控制系统，增益裕度一般为2～2.5，相位裕度为45°～65°。

本章小结

通过把 s 替代为 $j\omega$，我们可以实现 s 域和**频域**之间的转换。而**频率响应函数**是变换到频域内的传递函数。

系统的频率响应是当正弦输入信号在一定频率范围内变化时，得到的一系列的幅值 $|G(j\omega)|$ 和相角 ϕ 的变化值。这可以表示在两幅图中，一幅是幅值 $|G(j\omega)|$ 随角频率 ω 的变化图，另一幅是相角 ϕ 随角频率 ω 的变化图。幅值和角频率轴使用对数刻度。这两幅图称为**伯德图**。

我们可以通过把组成元素的增益曲线相加求得系统的伯德图。同样，相角曲线可以通过把组成元素的相角相加求得。

谐振峰值 Mp 定义为增益的最大值。带宽定义为增益不小于 -3dB 的频率带。在增益为 -3dB处的频率称为截止频率。

对于具有负反馈的自持振荡系统，即它处在**临界稳定状态**，系统的频率响应函数的增益是1，相角是 $-180°$。增益裕度是在相角是 $-180°$ 时增益比变为1需要乘以的系数，即处于临界不稳定处。相位裕度是增益是1时相角与 $-180°$ 相差的角度。

503

习题

21.1 具有以下传递函数的系统的增益和相角分别是多少？

(a) $\dfrac{5}{s+2}$ (b) $\dfrac{2}{s(s+1)}$ (c) $\dfrac{1}{(2s+1)(s^2+s+1)}$

21.2 在输入正弦信号 $3\sin(5t+30°)$ 的情况下，传递函数是 $1/(s+2)$ 的系统的稳态响应是什么？

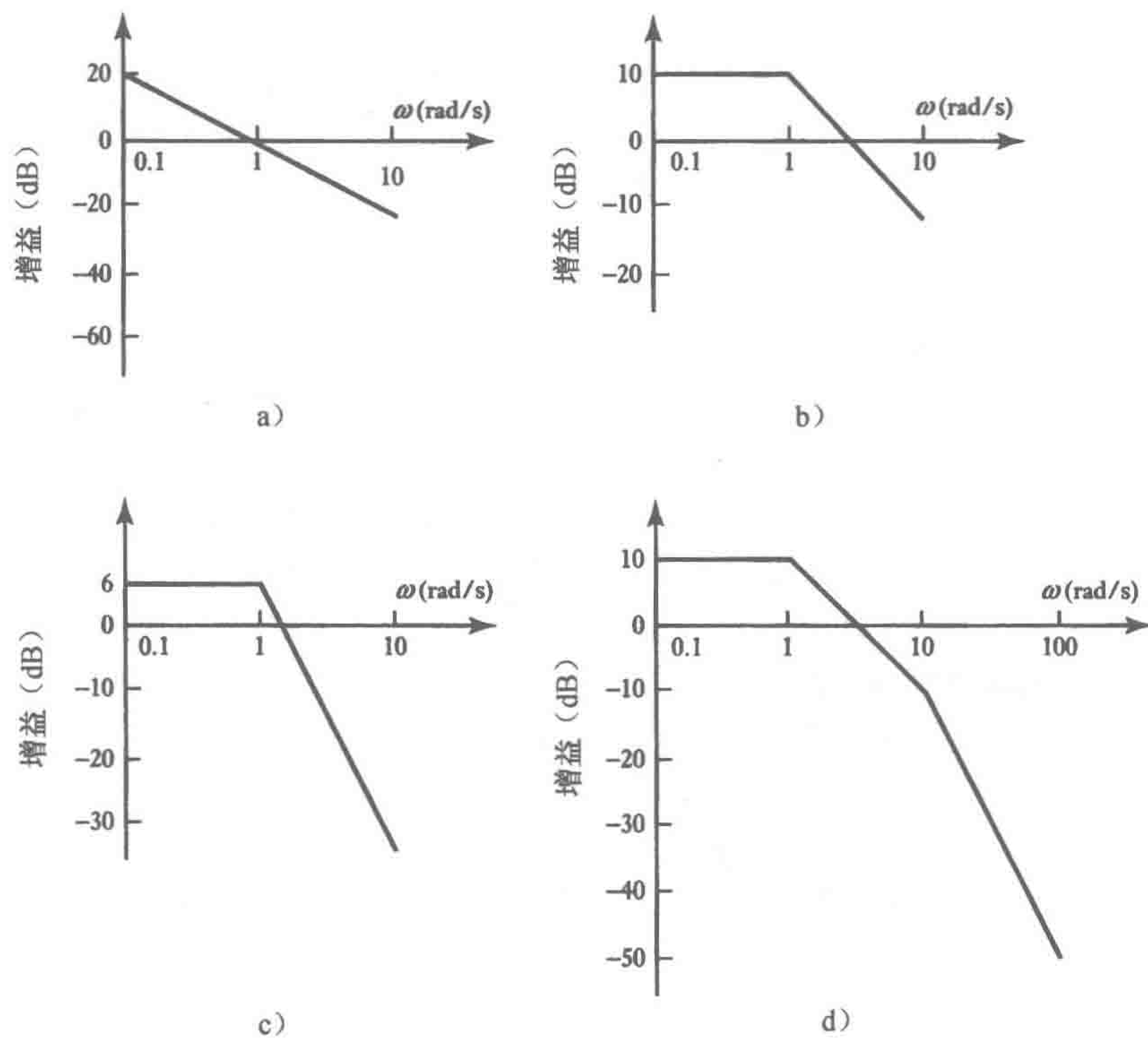
21.3 在输入正弦信号 $2\sin(2t+70°)$ 的情况下，传递函数是 $5(s^2+3s+10)$ 的系统的稳态响应是什么？

21.4 当传递函数是（a）$1/[s(2s+1)]$，（b）$1/(3s+1)$ 时，求得角频率分别为0rad/s，1rad/s，2rad/s，∞ rad/s时的增益和相角。

21.5 绘出传递函数是（a）$10/[s(0.1s+1)]$，（b）$1/[(2s+1)(0.5s+1)]$时系统的渐近伯

德图。

21.6 根据图 21-14 给定的伯德图求系统的传递函数。



504

图 21-14 习题 21.6

第22章　闭环控制系统

本章目的

通过本章的学习，读者将能够：

- 了解稳态误差。
- 了解两步控制（two-step control）模式的工作原理。
- 预测有比例、积分、导数、比例积分（PI）、比例导数（PD）以及比例积分导数（PID）控制的系统行为。
- 描述数字控制器的工作原理。
- 了解控制器参数调整方法。

22.1　连续与离散控制过程

开环控制是一种通常以“开-关”形式出现的控制，例如一个室内电加热器为了加热房间要么处于开启状态，要么处于关闭状态。**闭环控制系统**是一种控制器，它连续地将系统输出与期望值比较，并将误差转换成控制行为来消除误差。误差可能随着被控状态或设定值的改变而发生变化。例如，输入到系统的阶跃信号会改变系统的设定值为一个新值。本章我们将着重研究控制器如何随着误差信号的改变而做出相应的响应，即所谓的**控制模式**。例如，气动系统或运算放大器系统都可以采用这类控制器。但是，随着计算机系统的快速发展，它们将逐步被替代。**直接数字控制**是一种计算机在反馈回路和运动控制中采用的控制方式。本章主要介绍这一类闭环控制。

许多控制过程不仅对控制变量有要求（例如，温度需要达到期望值），而且也对操作顺序有要求。例如，家用洗衣机（见1.5.5节）需要按照预先设定好的顺序工作。另一个例子就是工业生产过程中控制系统按照既定顺序将诸多零件进行装配。操作顺序可分为：**基于时序的**操作顺序，**基于事件的**操作顺序，或者二者结合。在基于时序的系统中，系统在指定时间执行某个动作；在基于事件的系统中，系统在特定事件发生时执行某个动作。

在许多控制过程中，同时存在连续控制与离散控制两种模式。例如，家用洗衣机在一个洗涤周期内依序进行多个洗衣环节，并且通过反馈回路控制热水的温度和水位。

22.1.1　开环与闭环控制系统

闭环系统与开环系统的区别在于闭环系统具有反馈回路。开环系统的输入信号不会根据实际输出信号而做出调整。在闭环系统中，输出信号反馈并借此修正输入信号以使系统达到期望的输出。

反馈的一个结果是能够减弱干扰信号对系统的影响。干扰信号是不希望的信号，它能影响系统的输出。所有的物理系统都会受到各种形式的外来信号的干扰。例如，对于一台电动机，就存在电刷或者换向器噪声。

考虑外部扰动对整个开环系统增益的影响。图22-1展示了一个双元开环系统，在两个单元中间引入扰动。对于系统的参考输入 $R(s)$，第一个单元的输出为 $G_1(s)R(s)$。加入扰动 $D(s)$后，第二单元的输入信号为 $G_1(s)R(s)+D(s)$。整个系统输出 $X(s)$ 为：

$$X(s) = G_2(s)[G_1(s)R(s) + D(s)] = G_1(s)G_2(s)R(s) + G_2(s)D(s)$$

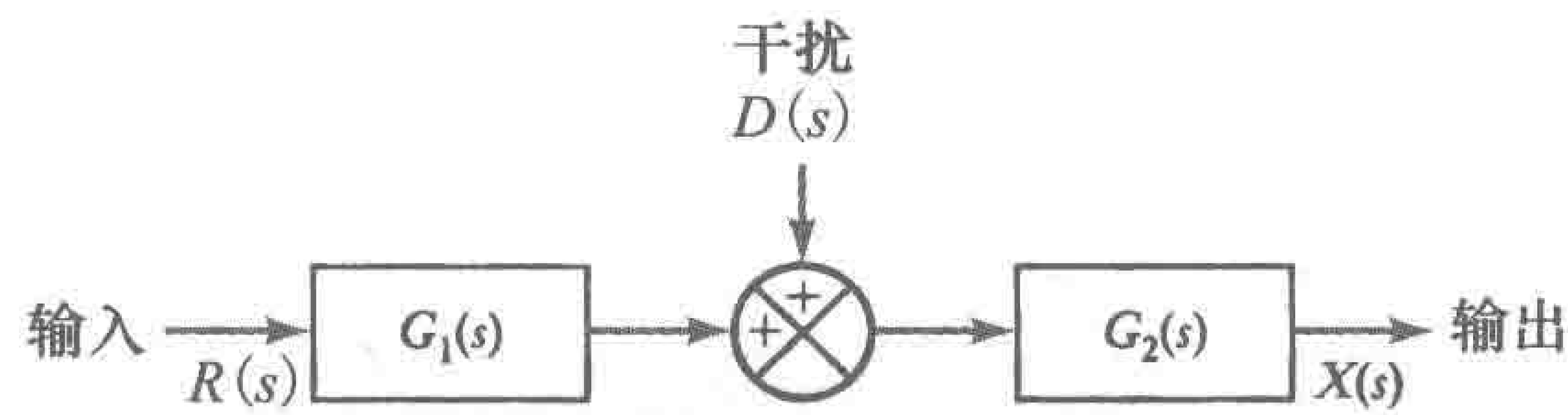


图 22-1　带干扰的开环系统

对于与之相对应的带负反馈的系统（见图 22-2），第一个前向通道单元 $G_1(s)$ 的输入信号为 $R(s) - H(s)X(s)$，所以它的输出信号为 $G_1(s)\ [R(s) - H(s)X(s)]$。$G_2(s)$ 的输入信号为 $G_1(s)\ [R(s) - H(s)X(s)] + D(s)$，所以它的输出信号为：

$$X(s) = G_2(s)\{G_1(s)[R(s) - H(s)X(s)] + D(s)\}$$

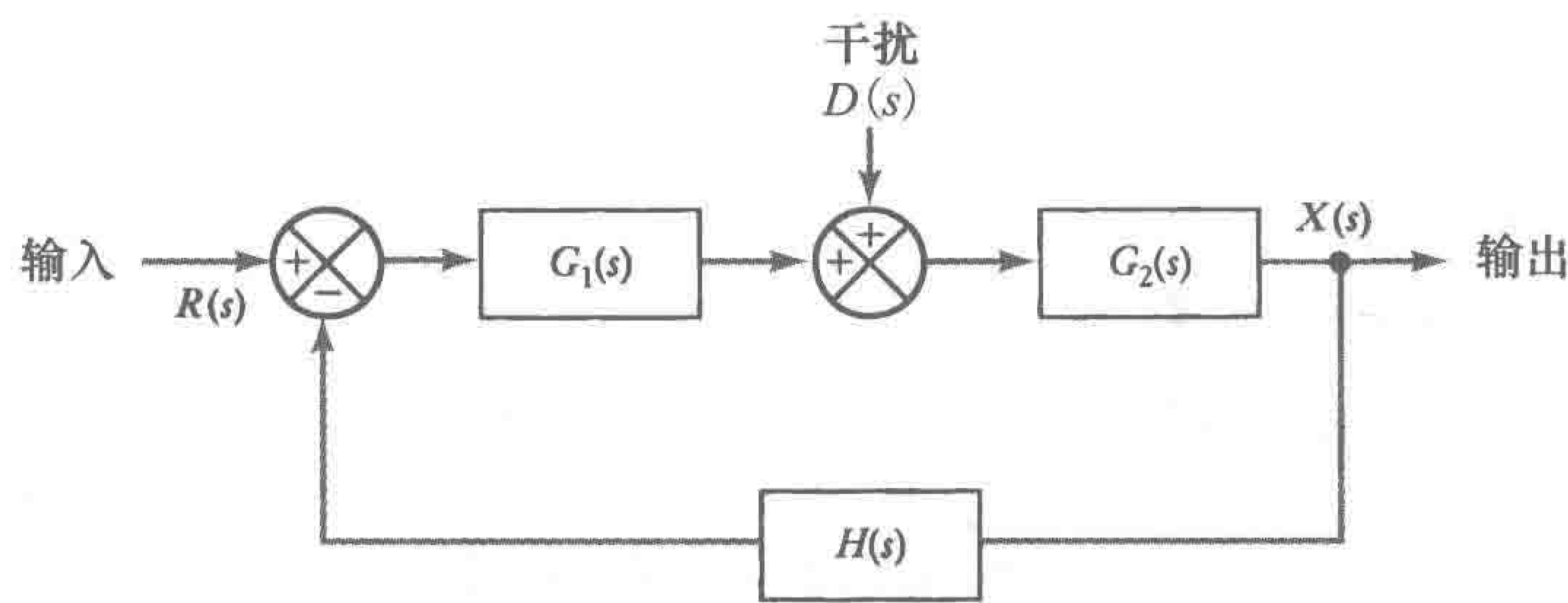


图 22-2　带干扰的闭环系统

因此，

506

$$X(s) = \frac{G_1(s)G_2(s)}{1 + G_1(s)G_2(s)H(s)}R(s) + \frac{G_2(s)}{1 + G_1(s)G_2(s)H(s)}D(s)$$

与开环系统方程相比较，我们可以看出闭环系统扰动对系统输出影响减小了 $[1 + G_1(s)G_2(s)H(s)]$ 倍。当引入反馈时，扰动的影响显著减少了。

22.2　术语

以下是闭环控制器的常用术语。

22.2.1　滞后

任何控制系统中都存在**滞后**（lag）。因此，被控条件的改变并不会立即产生来自控制系统的正确响应。这是因为系统需要时间来产生相应的响应。例如，中央空调系统对室内温度的控制。当房间温度低于设定温度时，控制系统开始打开加热器，这就产生了滞后。滞后并不是唯一的。即使当控制系统开始响应，室温的升高仍存在滞后，因为热量从加热器到房间空气的传递需要时间。

22.2.2　稳态误差

闭环控制系统将测量系统的输出值与期望值进行比较，进而产生误差信号。被控变量的改变或输入期望的变化都会产生误差信号，并被送入控制器中。例如，随时间递增的斜坡信号。当输入信号发生变化时，会产生瞬时响应。但是，瞬时响应会随时间减弱。**稳态误差**用来描述瞬态响应后期望值与稳态输出值的差值。因此，它是衡量控制系统跟踪输入值精度的一种方式。只要有误差存在，输出就不是期望的输出结果。

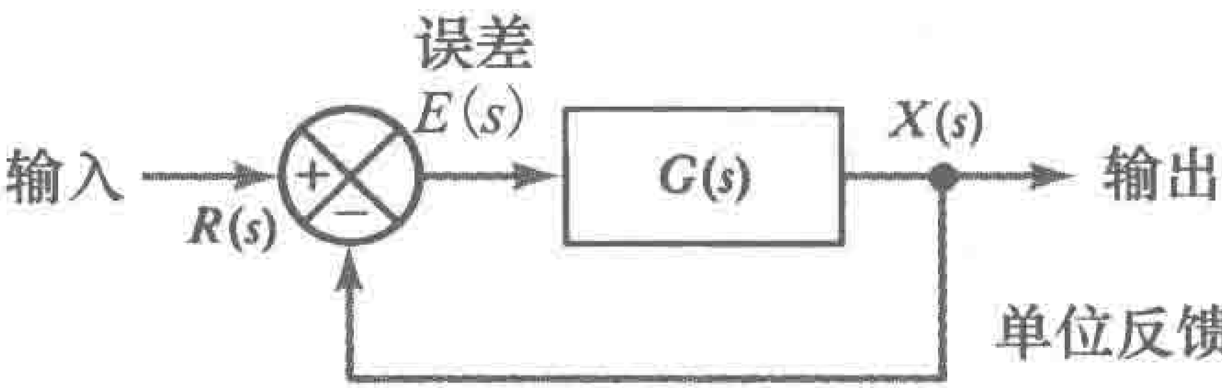


图 22-3　单位反馈

假设某一控制系统具有单位反馈（见图 22-3）。当参考输入信号为 $R(s)$ 时，输出为 $X(s)$。反馈信号是

$X(s)$，所以误差信号为 $E(s)=R(s)-X(s)$。如果 $G(s)$ 为前向通道的传递函数，那么单位反馈系统就有：

$$\frac{X(s)}{R(s)}=\frac{G(s)}{1+G(s)H(s)}=\frac{G(s)}{1+G(s)}$$

所以：

$$E(s)=R(s)-X(s)=R(s)-\frac{G(s)R(s)}{1+G(s)}=\frac{1}{1+G(s)}R(s)$$

因此，误差取决于 $G(s)$。

为了确定稳态误差值，假设误差 e 为时间的函数，然后当所有的瞬态响应都减弱时，时间函数 e 趋近于确定值。同时，我们能够求出 $E(s)$ 的反变换，进而得到当 $t\to\infty$ 时的值。通常我们采用一种简单的方法求解——**终值定理**（见附录 A）。这其中涉及当 s 趋近 0 时求 $sE(s)$ 的值：

$$e_{SS}=\lim_{t\to\infty}e(t)=\lim_{s\to 0}sE(s)$$

为了证明上式，设单位反馈系统的前向通道传递函数为 $k/(\tau s+1)$，输入为单位阶跃信号 $1/s$：

$$e_{SS}=\lim_{s\to 0}sE(s)=\lim_{s\to 0}\left[s\frac{1}{1+k/(\tau s+1)}\frac{1}{s}\right]=\frac{1}{1+k}$$

因此存在稳态误差。系统输出结果不会达到期望值。通过增加系统增益 k，系统稳态误差将会逐渐减小。

前向通道近似为一个带增益 k 的控制器和传递函数为 $1/(\tau s+1)$ 的系统，这样的控制器称为比例控制器。在这种情况下，稳态误差通常称为**偏移量**。通过增加增益可逐渐缩小误差。

但是，如果单位反馈系统有一个传递函数为 $k/s(\tau s+1)$ 的前向通道，那么对于阶跃信号，系统的稳态误差为：

$$e_{SS}=\lim_{s\to 0}sE(s)=\lim_{s\to 0}\left[s\frac{1}{1+k/s(\tau s+1)}\frac{1}{s}\right]=0$$

系统不存在稳态误差。在这类系统中，前向通道是一个具有增益 k/s 的控制器和传递函数为 $1/(\tau s+1)$ 的系统。这样的控制器没有偏移量，称为积分控制器。通过组合积分和比例控制器，可以有效地消除偏移量。如果增加一个微分控制器，就能使控制系统对变化产生更迅速的响应。

507～508

22.2.3　控制模式

控制单元可以通过多种模式响应误差信号，并为校正单元提供输出。

1）两步模式：控制器本质上是一个开关，由误差信号激励，打开或关闭（on/off）校正信号。

2）比例模式（P）：该模式产生的控制动作与误差成比例。误差信号越大，校正信号就越大。校正信号随着误差信号减弱而减弱，同时校正过程也逐渐减慢。

3）微分模式（D）：该模式产生的控制动作与误差变化率成正比。当误差信号突然改变时，控制器给出一个很大的校正信号；当误差逐渐改变时，只有很小的校正信号产生。微分控制被认为是一种超前控制，通过测量实时的误差变化率，控制器能够预测较大的误差，并在较大误差出现前执行校正。微分控制不能单独使用，常与比例控制或积分控制组合。

4）积分模式（I）：该模式产生的控制动作与误差随时间的积分成正比。因此固定的误差信号将会产生递增的校正信号。只要误差存在，校正就会持续增加。积分控制器对误差求和，并根据已发生的变化进行响应，可认为是对历史的“回顾”。

5）组合模式：比例微分（PD）模式、比例积分（PI）模式、比例积分微分（PID）模

式。术语**三项控制器**用于 PID 控制。

以上五种控制模式在下面各节中进行讨论。控制器通过气动回路、带运算放大器的模拟电路或者微处理器、计算机编程来实现这些控制模式。

22.3　两步控制模式

两步控制模式的典型案例就是双金属温度调节器（见图 2-46），常用于简单的温度控制系统。它是一个感温开关。如果室温高于设定温度时，双金属片就处于关闭位置，加热器关闭；如果室温低于设定温度时，双金属片就变换开关位置，加热器完全打开。这种形式的控制器只有两个位置，打开或者关闭，如图 22-4a 所示。

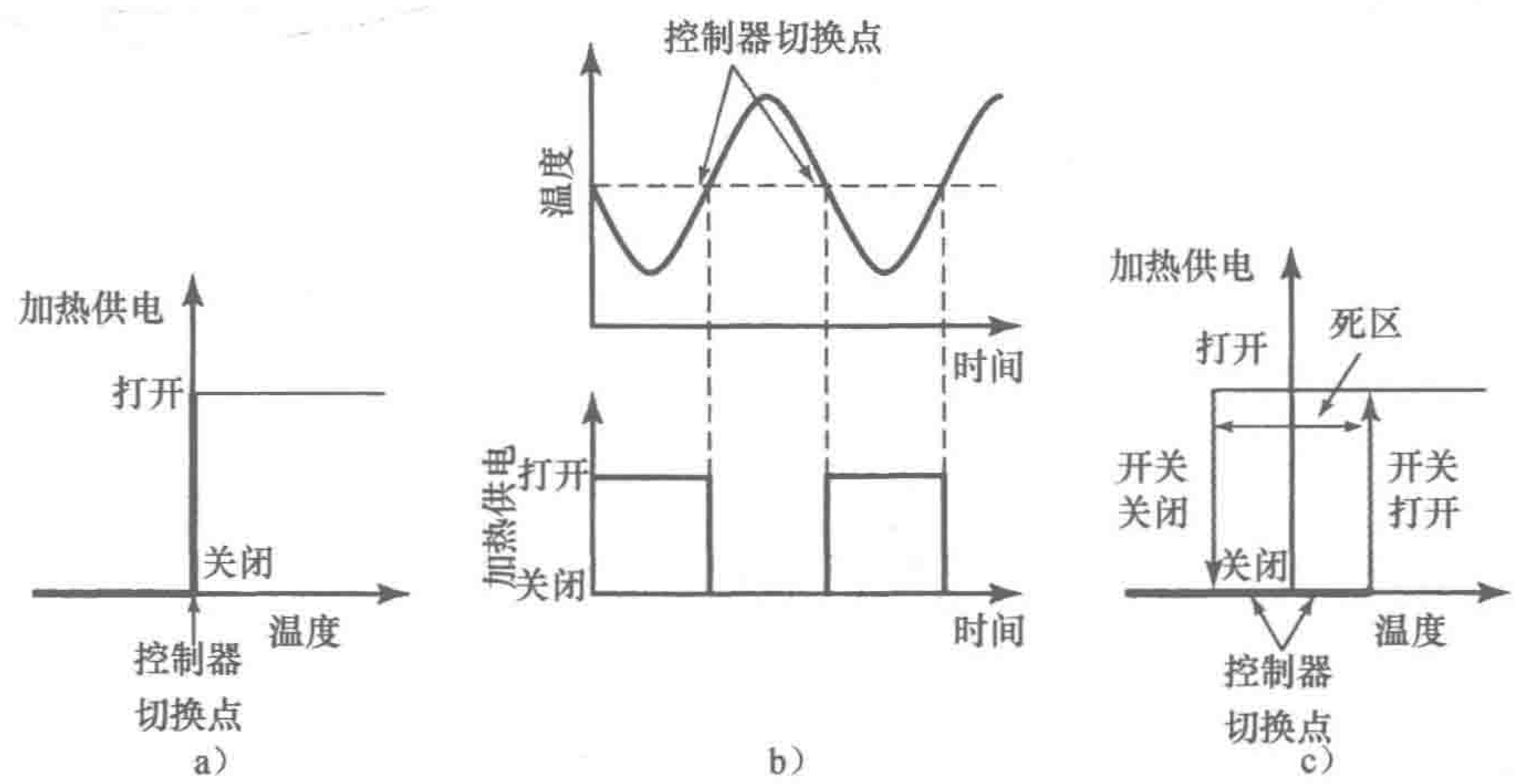


图 22-4　两步控制

两步控制模式的控制动作是不连续的。这就导致了被控变量在期望值附近振荡。因为时滞的存在，所以控制系统和过程需要时间来响应。以家用中央空调的温度控制为例，当房间温度下降到设定值以下时，从控制系统响应到打开加热器所消耗的时间，比加热器对房间温度开始产生影响前的时间更短。在此期间，温度下降更多。相反的现象也会发生在房间温度上升到设定温度以上时。由于控制系统发生反应并关闭加热器需要时间，而且加热器冷却并停止加热房间也需要更多的时间，所以房间温度超过了期望值。这就导致了房间温度在设定温度上下振荡（见图 22-4b）。

对于上述简单的两步系统来说存在一个问题，即当房间温度在设定值附近振荡时，恒温器可能不停地打开或关闭以响应非常细微的温度变化。因此，为了避免这种情况，采用高、低双温度值代替单温度值，在温度低于低温度值时加热器打开，在温度高于高温度值时加热器关闭，从而避免在一个温度点反复启动或者关闭加热器（见图 22-4c）。术语**死区**用于表示开、关加热器的高、低温度值之间的温度区间。死区较大将会导致温度在设定值附近波动很大；死区较小将会提高开、关的频率。如图 2-46 所示，双金属元件有一个永磁的开关触点，这就产生了死区的效果。

两步控制常用于变化缓慢的控制中，即一个具有较大容抗的过程中。以房间加热为例，开、关加热器对房间温度的作用都是缓慢变化的。这就产生了一个周期较长的振荡过程。两步控制精度不高，但适用于结构简单的设备，而且成本低廉。开/关控制不局限于机械开关，如双金属片或者延时器；快速开关可以用晶体管电路（详见 9.3.2 节）实现。晶体管电路可用于控制电动机和运算放大器的速度。

22.4 比例模式

509~510

不论误差大小，采用两步控制模式的控制器输出非开即关。而在比例模式中，控制器输出大小与误差大小成正比。误差越大，控制器的输出信号值越大。这就意味着控制系统的校正元件，例如阀门，将会收到一个与要求的校正大小成正比的信号。因此，

$$控制器输出 = K_P e$$

其中 e 是误差，K_P 为常量。采用拉普拉斯变换，

$$控制器输出(s) = K_P E(s)$$

所以 K_P 是控制器的传递函数。

22.4.1 电子比例控制器

一个带反相运算功能的加法运算放大器可用做一个比例控制器（见图22-5）。对于加法放大器，有（见3.2.3节）：

$$V_{out} = -R_f\left(\frac{V_0}{R_2} + \frac{V_e}{R_1}\right)$$

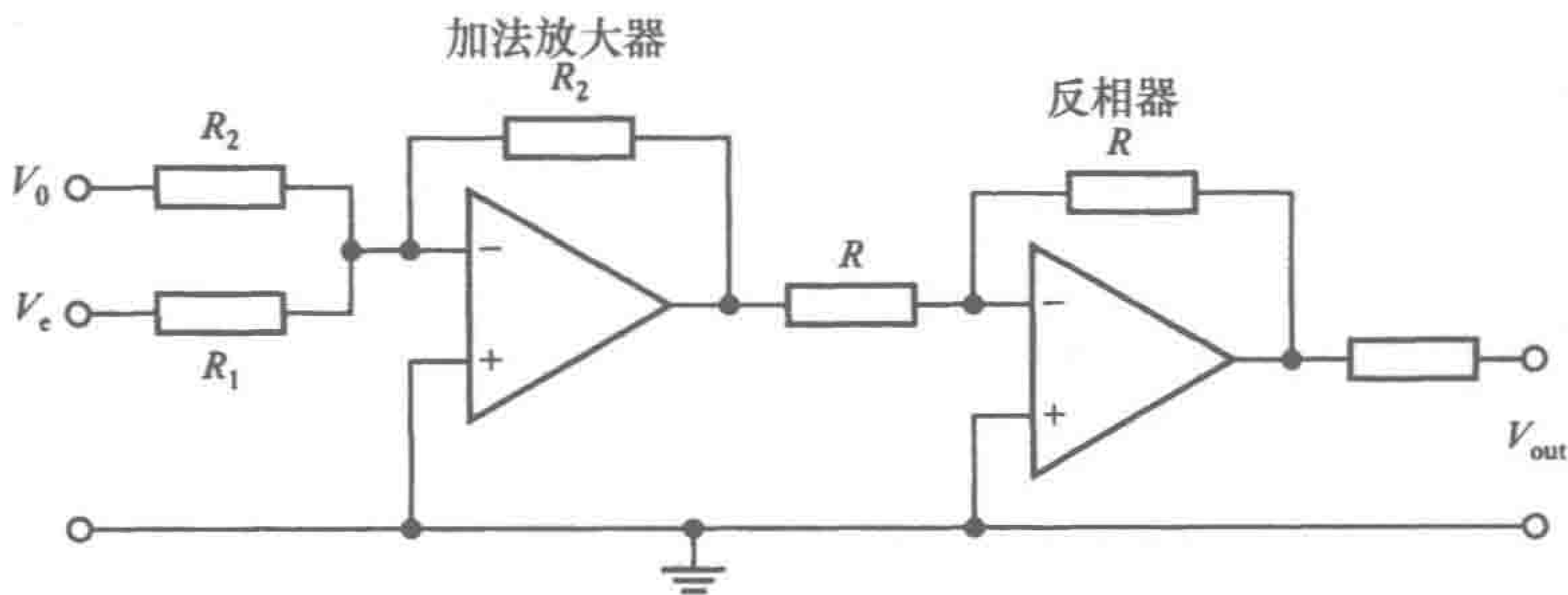


图22-5 比例控制器

经过 R_2 的加法放大器的输入信号是一个零误差电压值 V_0，即设定值，和经过 R_1 的误差输入信号 V_e。当反馈电阻 $R_f = R_2$ 时，方程变为：

$$V_{out} = -\frac{R_2}{R_1}V_e - V_0$$

如果加法放大器的输出信号通过反相器，即反馈电阻等于输入电阻的运算放大器，那么：

$$V_{out} = \frac{R_2}{R_1}V_e + V_0$$

$$V_{out} = K_P V_e + V_0$$

其中 K_P 是比例常量。该结果就是比例控制器。

举例说明，图22-6表示一个控制容器内液体温度的比例控制系统，液体通过该装置被抽出。 511

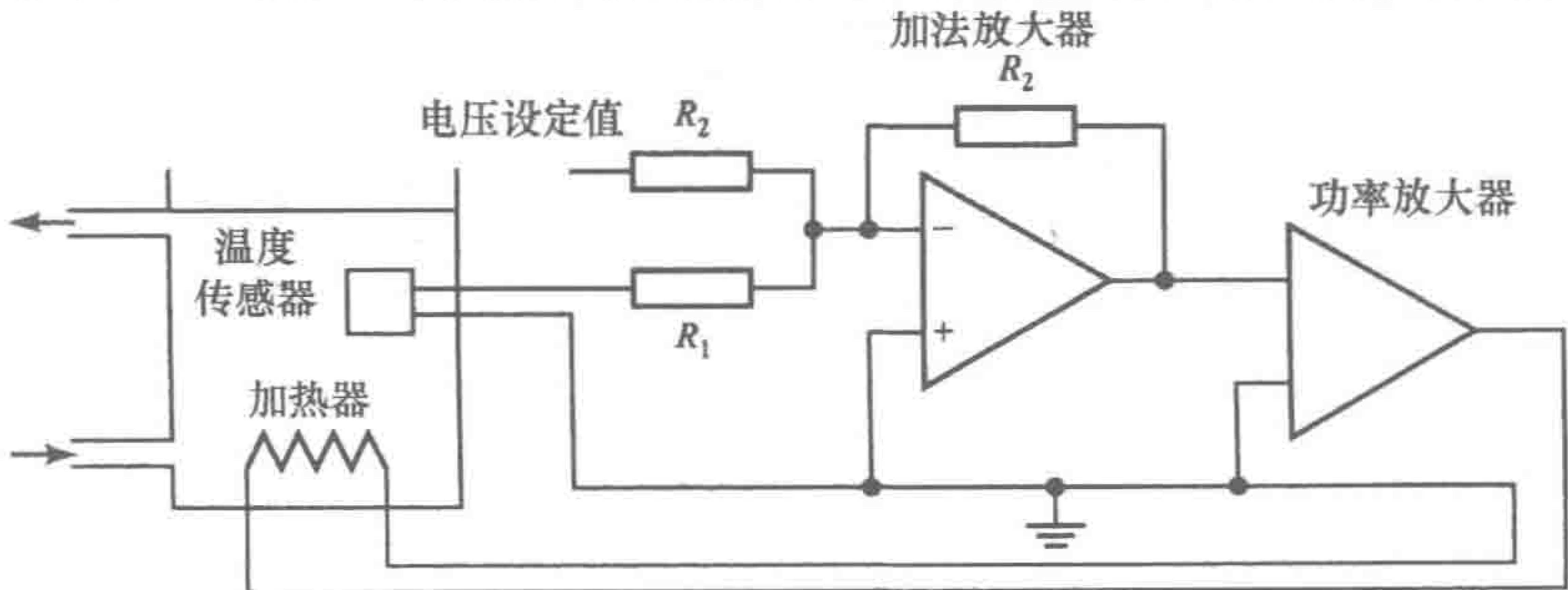


图22-6 温度控制的比例控制器

22.4.2 系统响应

在比例控制中，一个传递函数为 K_P 的增益元件与前向通道元件 $G(s)$ 串联（见图 22-7）。误差为：

$$E(s) = \frac{1}{1 + K_P G(s)} R(s)$$

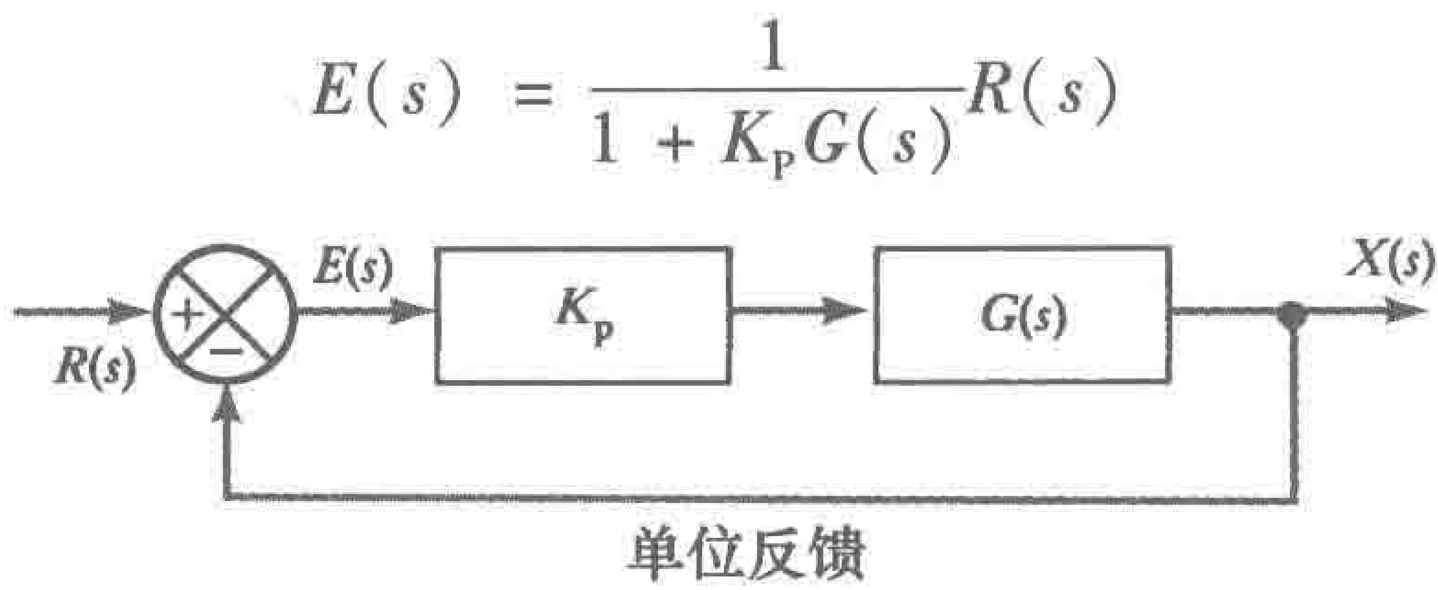


图 22-7 比例控制系统

所以，对于阶跃输入信号，系统稳态误差为：

$$e_{SS} = \lim_{s\to 0} sE(s) = \lim_{s\to 0}\left[s\frac{1}{1 + K_P G(s)}\frac{1}{s}\right]$$

这是一个有限值，所以稳态误差一直存在。较小的 K_P 值会产生很大的稳态误差，但会得到稳定响应。较大的 K_P 值会有较小的稳态误差，但有较大的不稳定趋势。

22.5 微分控制

采用**微分控制模式**的控制器的输出与误差信号对时间的变化率成正比。用方程可以表示为：

$$控制器输出 = K_D \frac{de}{dt}$$

其中 K_D 是一个比例常量。传递函数通过拉普拉斯变换可得：

$$控制器输出\ (s) = K_D sE(s)$$

因此传递函数为 $K_D s$。

512 具有微分控制的系统中，一旦误差信号发生变化，就会产生一个与误差信号变化率成正比的较大的输出信号，而不是误差信号值本身。因此，对误差信号的快速初始响应就出现了。如图 22-8 所示为当误差信号以均匀速率变化时控制器的输出结果。控制器的输出信号是伴随着误差信号的出现而产生的，而且，因为误差变化率是常量，所以控制器的输出也是常量。但是，微分控制器对稳态误差信号是没有响应的，这是因为稳态误差随时间的变化率为零。由于上述原因，微分控制器与比例控制器一起使用，比例控制部分对所有误差都能够产生响应，其中包括稳态误差，而微分控制部分只对变化率产生响应。微分控制也存在一个问题，如果过程变量的测量过程有噪声，噪声的快速波动会导致输出信号的产生，控制器会将该输出信号视为快速变化的误差信号，从而引起控制器的输出信号显著提高。

图 22-8 微分控制

图 22-9 所示为电子微分控制器的电路形式，其中包括含有一个作为微分电路的运算放大器，后面有一个作为反相器的运算放大器。微分时间 K_D 等于 R_2C。

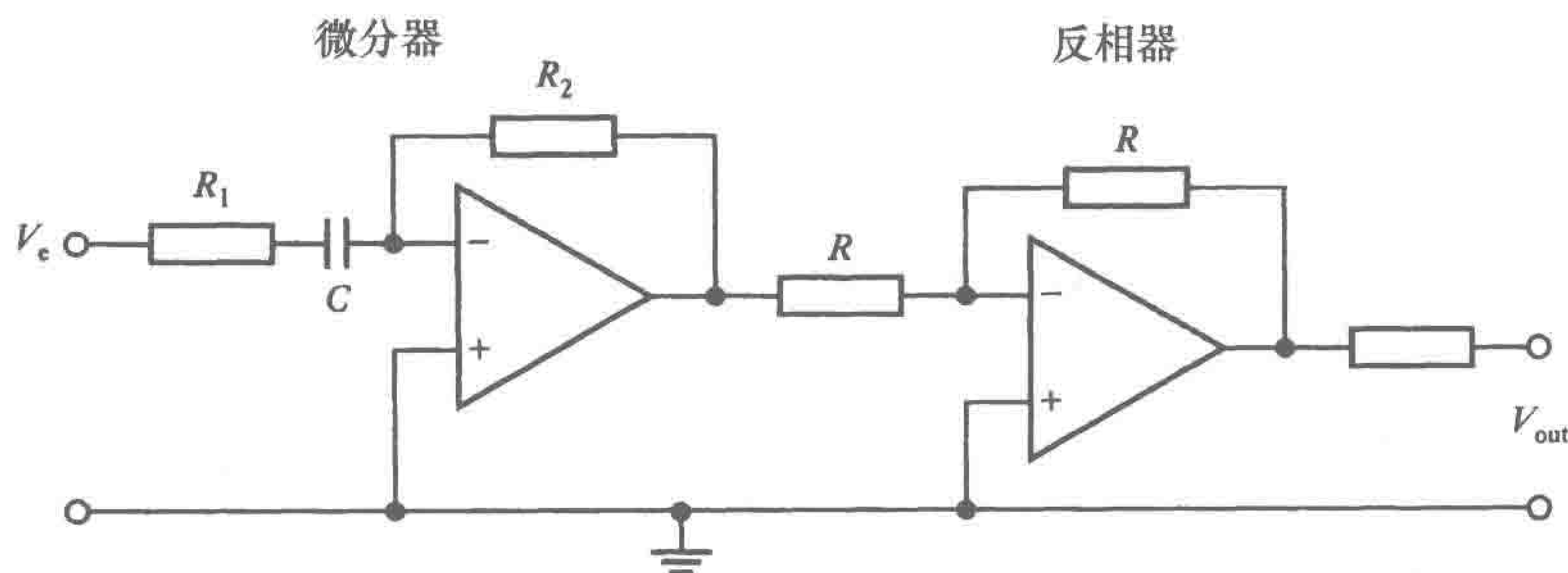


图 22-9　微分控制器

22.5.1　比例微分（PD）控制

因为微分控制在误差稳定时不会产生输出信号，控制器无法校正，所以微分控制不会单独使用。为了解决这个问题，我们常常将其与比例控制配合使用。

带比例微分控制的控制器输出为：

$$\text{控制器输出} = K_P e + K_D \frac{\mathrm{d}e}{\mathrm{d}t}$$

K_P 是比例常量，K_D 是微分常量，$\mathrm{d}e/\mathrm{d}t$ 是误差变化率。系统传递函数为：

$$\text{控制器输出}(s) = K_P E(s) + K_D sE(s)$$

因此传递函数是 $K_P + K_D s$。这经常可以写为：

$$\text{传递函数} = K_D\left(s + \frac{1}{T_D}\right)$$

其中 $T_D = K_D/K_P$，它也称为**微分时间常量**。

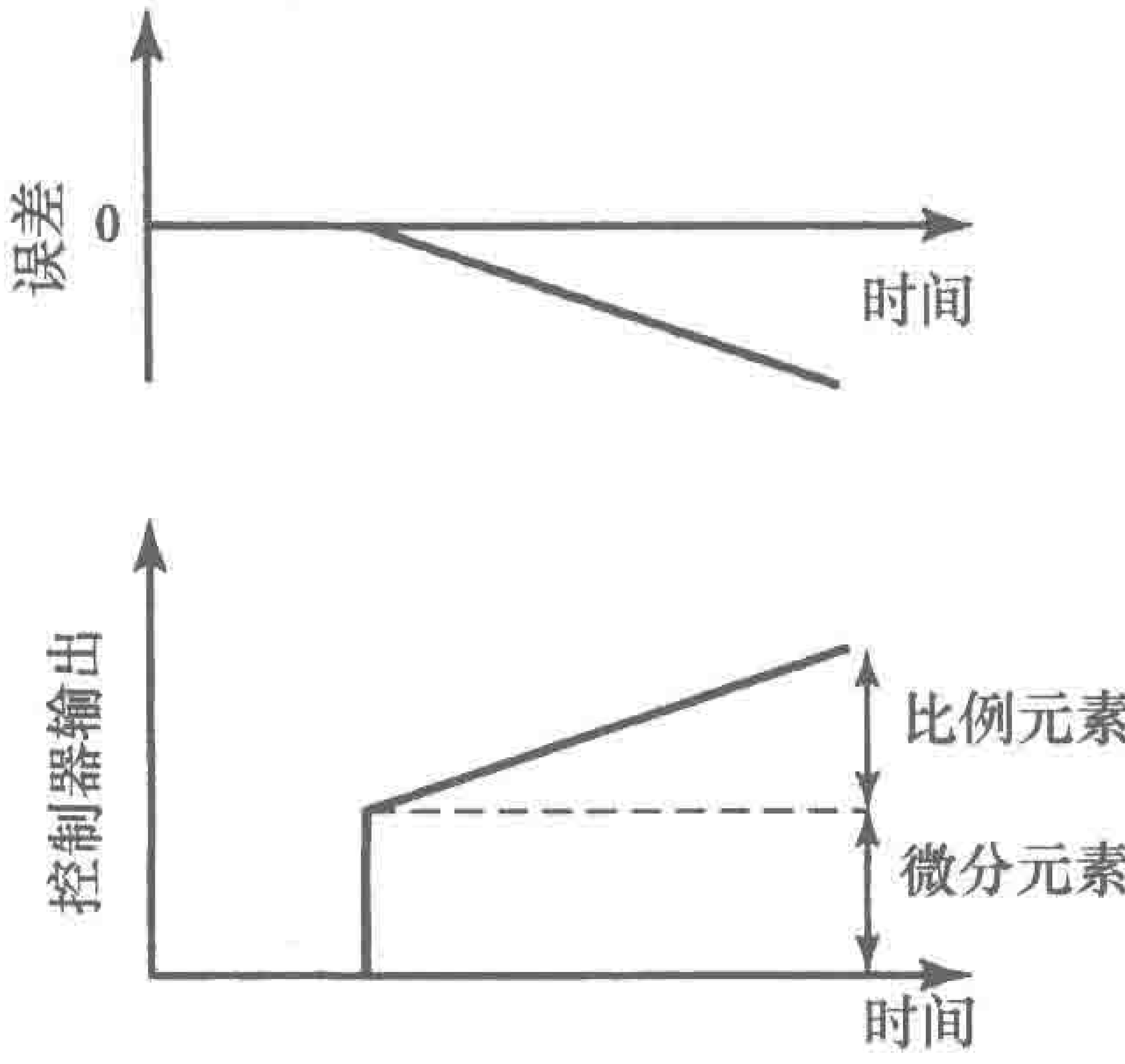


图 22-10　*PD* 控制

513

如图 22-10 给出了控制器的输出信号是如何随误差信号的不断改变而发生改变的。由于控制器具有微分控制功能，所以其初始输出信号能够快速变化，又由于具有比例控制功能，所以控制输出信号会逐渐发生变化。因此，这种形式的控制能够处理快速变化过程。

22.6　积分控制

积分控制模式的控制输出 I 的变化率与输入误差信号 e 成正比：

$$\frac{\mathrm{d}I}{\mathrm{d}t} = K_I e$$

K_I 为比例常量，单位为 $1/s$。对上式积分，得到：

$$\int_{I_0}^{I_{out}} \mathrm{d}I = \int_0^t K_I e \mathrm{d}t$$

$$I_{out} - I_0 = \int_0^t K_I e \mathrm{d}t$$

I_0 是控制器在零时刻的输出，I_{out} 为 t 时刻的输出。

传递函数经拉普拉斯变换后，得到：

$$(I_{out} - I_0)(s) = \frac{1}{s} K_I E(s)$$

所以，

$$传递函数 = K_I/s$$

图 22-11 所示为当有固定误差输入时积分控制器的输出。我们可以从两方面分析这个图。当误差为零时，控制器的输出信号为常量；当误差为常量时，控制器的输出信号以均匀速率变化。另一种研究该图的方法是计算误差 - 时间曲线下的面积：

514

$$从时间 0 \sim t 的误差曲线下的面积 = \int_0^t e\mathrm{d}t$$

在误差出现之前积分值始终为 0，因此 $I_{out} = I_0$。当误差出现时，误差信号保持为常量。因此输出曲线下的积分面积随时间递增。因为积分面积以匀速增长，所以控制器的输出信号也匀速增长。

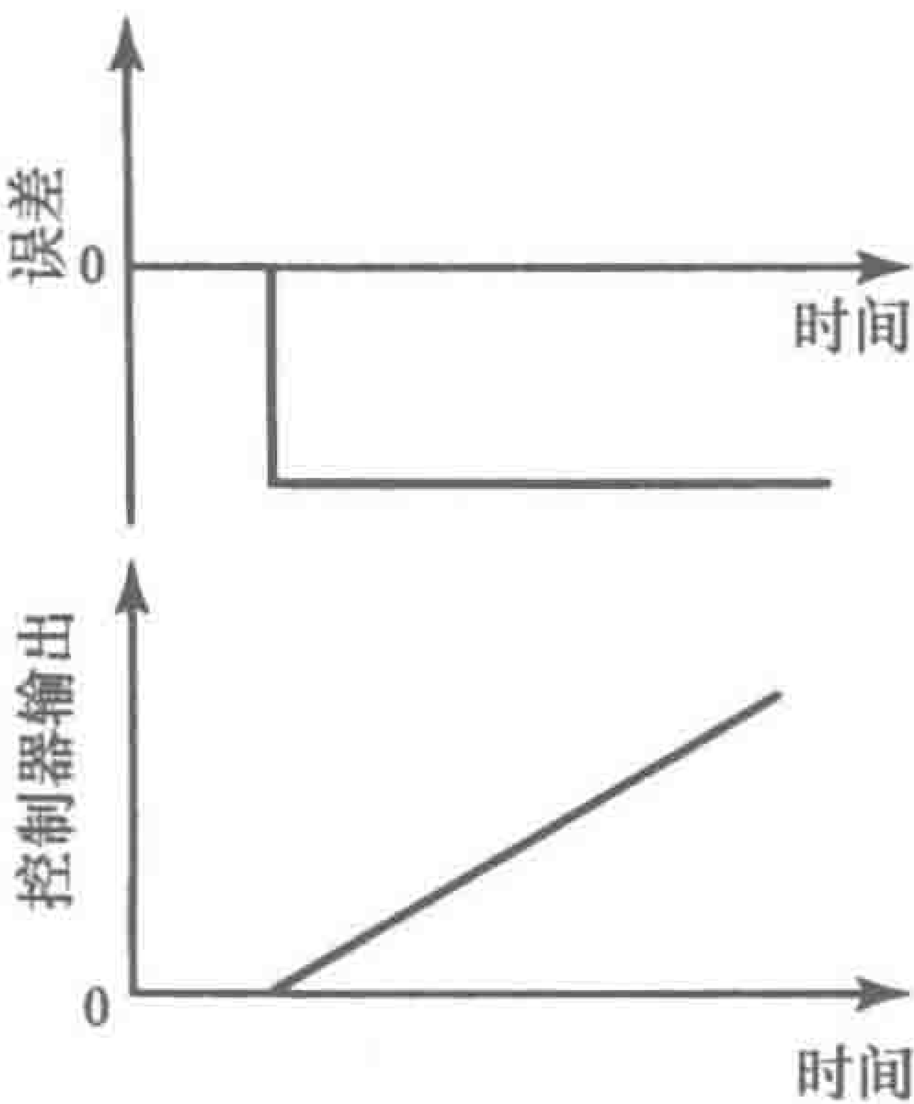


图 22-11　积分控制

图 22-12 所示为电子积分控制器的电路形式。它包括积分运算放大器和加法运算放大器，把零时刻控制器的输出与积分器的输出相加。K_I 等于 $1/R_1C$。

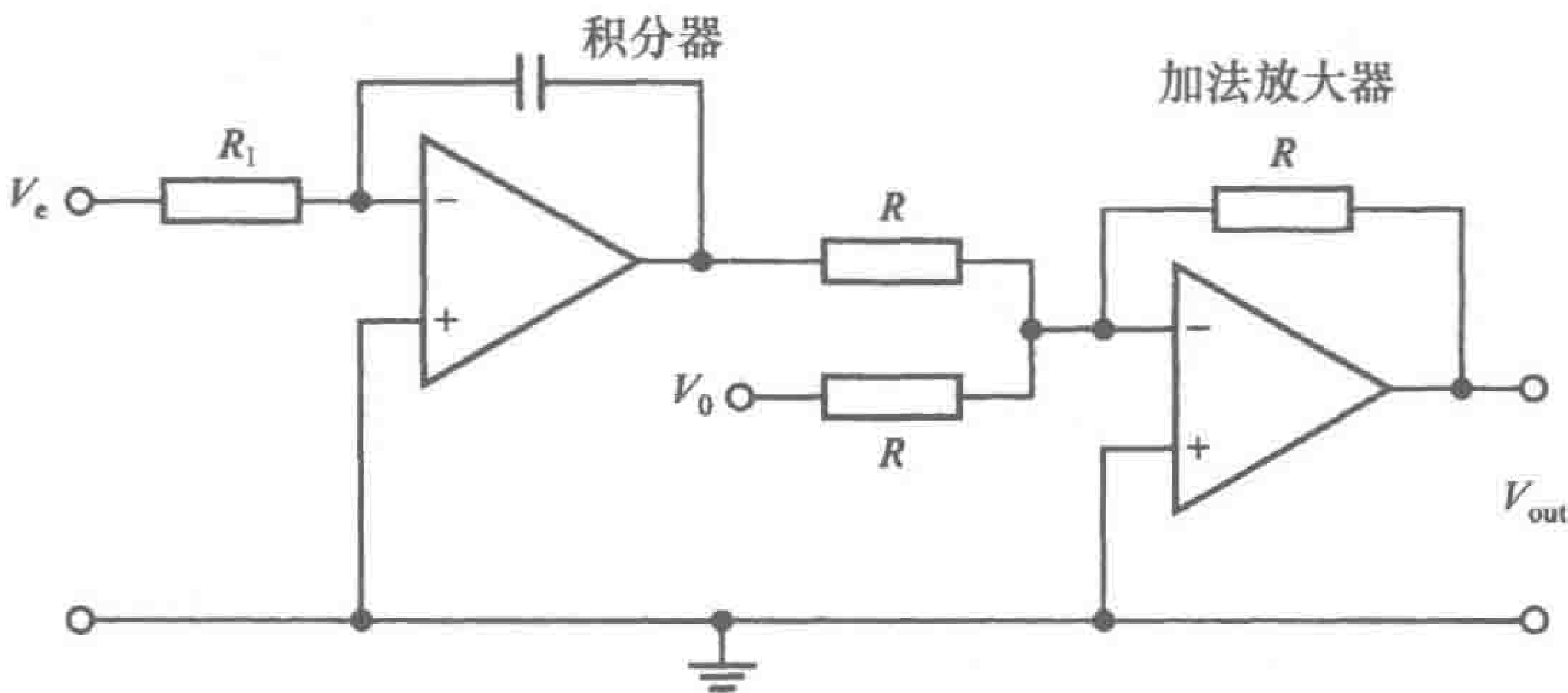


图 22-12　积分控制器

22.6.1　比例积分（PI）控制

积分控制模式通常不会单独使用，常与比例模式配合使用。当积分控制加入比例控制系统中时，控制器输出为：

$$控制器输出 = K_pe + K_I\int e\mathrm{d}t$$

其中 K_P 为比例控制常量，K_I 为积分控制常量，e 为误差。所以传递函数为：

$$传递函数 = K_P + \frac{K_I}{s} = \frac{K_P}{s}\left(s + \frac{1}{T_I}\right)$$

其中 $T_I = K_P/K_I$ 为**积分时间常量**。

图 22-13a 所示为当有一个突变为常量误差时的系统响应情况。误差使比例控制器输出信号升高，该输出信号在误差不变时保持稳定。由于积分控制的作用，系统还叠加了一个稳定增加的控制器输出信号。图 22-13b 所示为一个从零值递增然后又递减的误差信号，并给出了比例作用与积分作用叠加的效果。单独的比例控制器反映了误差的变化，并且最终回到了它的原始设定值。因为具有积分控制作用，所以在误差 - 时间曲线上控制器的输出信号值与曲线下的面积成正比，即使误差值回到 0，仍然存在一个面积值，误差终止后控制器仍保持输出。

515

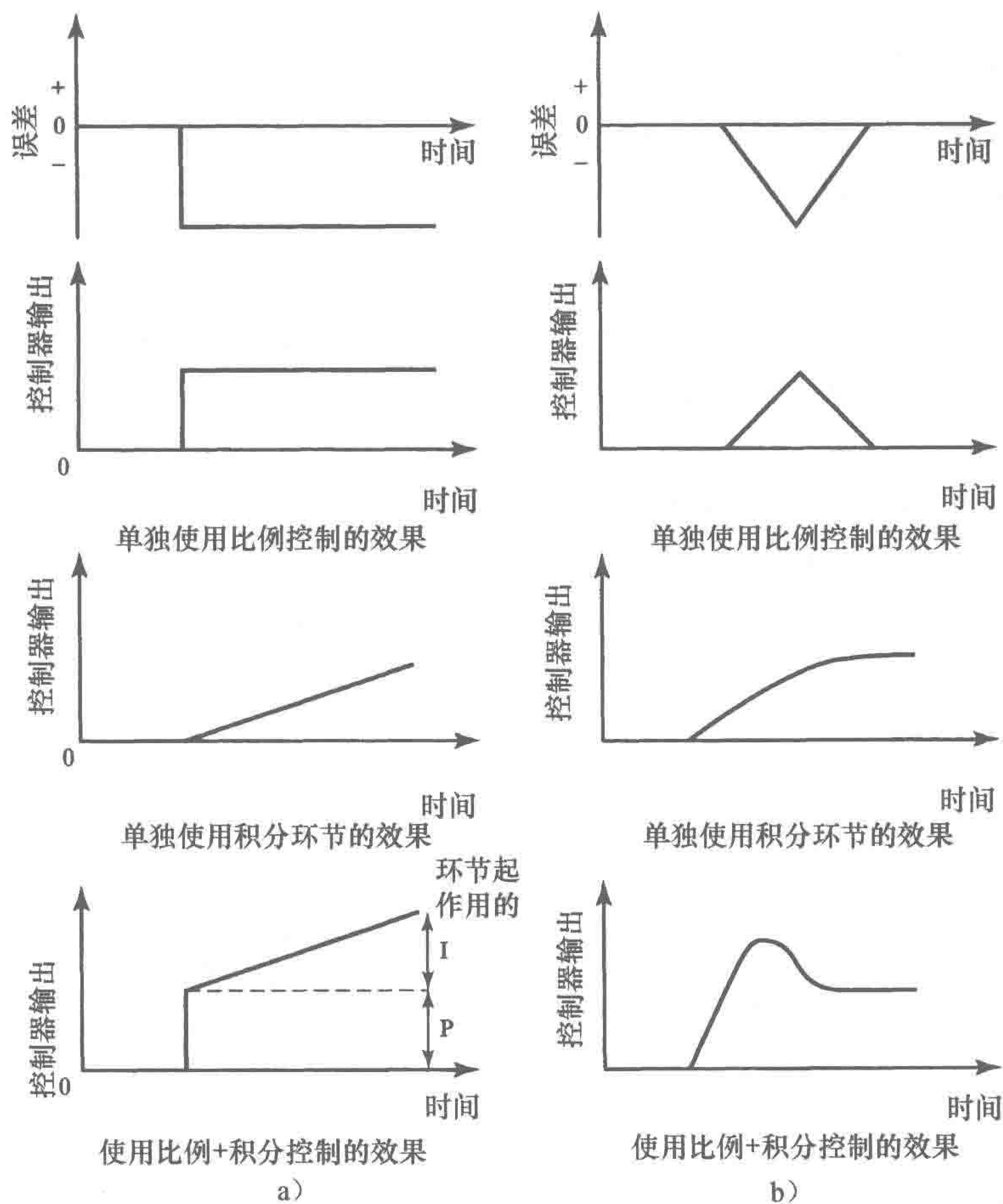


图 22-13 PI 控制

22.7 PID 控制器

三种控制模式（比例、积分和微分）结合形成的控制器称为**三项控制器**或者 **PID 控制器**。可用如下方程描述其作用：

$$\text{控制器输出} = K_P e + K_I \int e\mathrm{d}t + K_D \frac{\mathrm{d}e}{\mathrm{d}t}$$

其中 K_P 是比例常量，K_I 为积分常量，K_D 为微分常量。经拉普拉斯变换后有：

$$\text{控制器输出}(s) = K_P E(s) + \frac{1}{s} K_I E(s) + sK_D E(s)$$

所以

$$\text{传递函数} = K_P + \frac{1}{s} K_I + sK_D = K_P\left(1 + \frac{1}{T_I s} + T_D s\right)$$

516

22.7.1 运算放大器 PID 电路

通过组合本章之前所描述的比例、微分和积分电路，可以得到一个三项控制器。但是，我们也可以用一个运算放大器得到一个更实用的控制器，如图 22-14 所示。比例常量 $K_P = R_I/(R + R_D)$，微分常量 $K_D = R_D C_D$，积分常量 $K_I = 1/R_I C_I$。

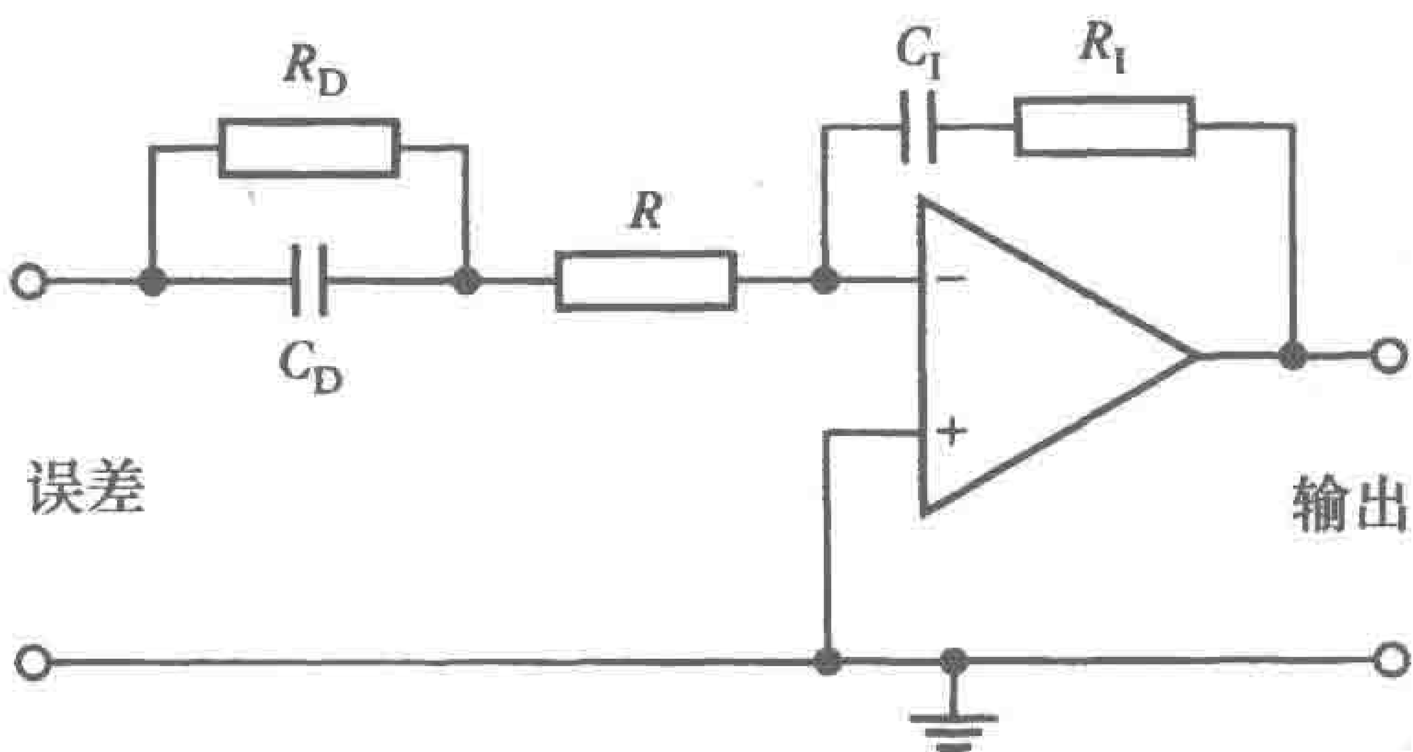
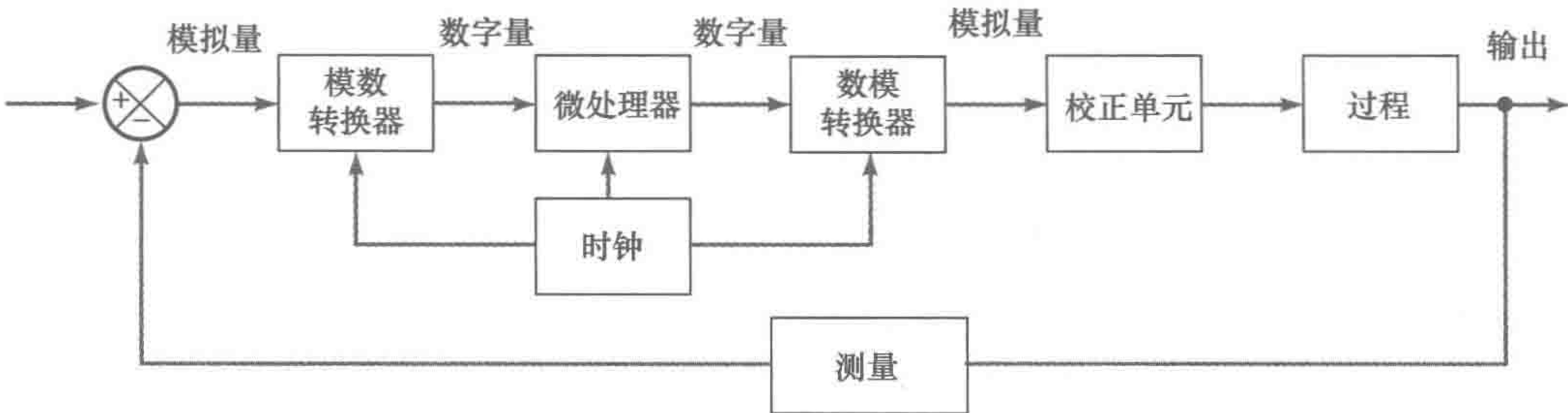


图 22-14 PID 电路

22.8 数字控制器

图22-15给出了一个可用于连续过程的直接数字控制系统的基本原理。术语**直接数字控制**应用于微处理器这样的数字控制器中，用于控制闭环系统。控制器接收传感器信号，执行控制程序，向校正单元输出信号。这类控制器要求输入信号为数字信号，并以数字形式对信息进行处理，且输出数字信号。因为许多控制系统都具有模拟测量功能，所以我们将模数转换器用于信号输入。时钟提供固定时间间隔的脉冲触发模数转换器对受控变量定时采样。将采样值转换成数字信号，与微处理器中的设定值相比较产生误差信号。微处理器采用一种控制模式来处理误差信号，并产生输出。微处理器使用的控制模式由微处理器指令程序（即软件）确定。因为校正机构需要模拟信号，所以数字输出信号通常需要经数模转换后，才能开始校正。



517

图22-15　数字闭环控制系统

数字控制器通常循环执行以下事件周期：

1）对测量数值进行采样。

2）采样值与设定值比较并计算误差。

3）基于误差值、先前存储的输入和输出值进行计算，获得输出信号。

4）发送输出信号到数模转换器。

5）在重复周期之前等待，直到下一个采样时间。

微处理器作为控制器比传统模拟形式的控制器更具优势。例如，比例控制模式或者三项控制模式都可以通过改变程序来实现，而不需要改变硬件或电路。为了使控制策略适应变化的环境，在控制行为中控制策略可以通过软件来改变。

它还具有其他优势。在模拟控制中，每一个被控过程都需要不同的控制器。而在微处理器中不同的过程可以采用一个多路复用器进行多路采样从而实现控制（见4.4节）。数字控制比模拟控制的精度更高，因为模拟系统的放大器和其他电子元器件的特性会随时间和温度发生改变，因而表现出漂移，而数字控制仅仅是以开/关模式进行信号的操作，不会发生这样的漂移。

22.8.1 控制模式的实现

数字控制器为了实现特定控制模式，必须编写一个合适的程序。为了使后面的校正单元获得期望输出，程序必须指明某一时刻的数字误差信号应该如何处理。处理过程将当前输入和前一个输入、输出联系起来，然后程序请求控制器执行差分方程（见4.6节）。

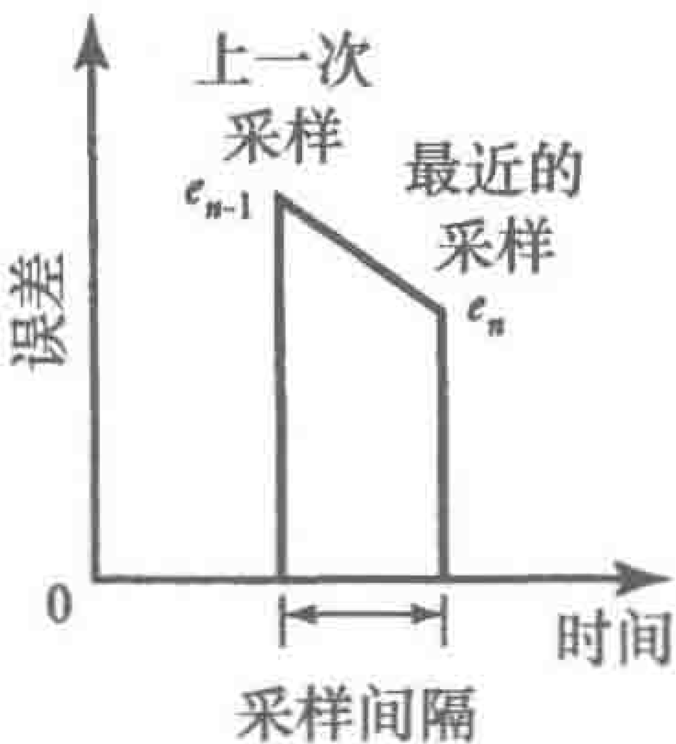


图22-16　误差信号

PID模拟控制器的传递函数是：

$$传递函数 = K_P + \frac{1}{s}K_I + sK_D$$

与 s 相乘表示微分。我们可考虑当前瞬时误差信号在时间响应上的梯

度等于（最新的误差采样 e_n 减去上一次误差采样 e_{n-1}）/（采样间隔 T_S）（见图 22-16）。

除以 s 表示积分。在一个采样周期结束时，误差的积分等于上一次采样间隔中误差 - 时间曲线中的面积加上之前所有采样面积的总和（Int_{prev}）。如果采样周期很短，那么上一次采样间隔的面积大约为$\frac{1}{2}(e_n+e_{n-1})/T_S$（见 4.6 节塔斯廷逼近）。因此，我们可以得到某一瞬时控制器输出 x_n 的传递函数的等效形式：

518

$$x_n = K_P e_n + K_I\left(\frac{(e_n+e_{n-1})T_s}{2} + \text{Int}_{\text{prev}}\right) + K_D\frac{e_n - e_{n-1}}{T_s}$$

等式简化为：

$$x_n = Ae_n + Be_{n-1} + C(\text{Int}_{\text{prev}})$$

其中 $A=K_P+0.5K_IT_s+K_D/T_s$，$B=0.5K_IT_s-K_D/T_S$ 和 $C=K_I$。

于是，PID 控制器的程序变为：

1）设定 K_P、K_I 以及 K_D 的值。

2）设定 e_{n-1}、Int_{prev}以及采样时间 T_S 的初始值。

3）复位采样间隔定时器。

4）输入误差 e_n。

5）通过上式计算 y_n。

6）为了下一次计算，将前一面积值更新为 $\text{Int}_{\text{prev}}+0.5(e_n+e_{n-1})T_S$。

7）为了下一次计算，更新误差值 e_{n-1}等于 e_n。

8）等待采样间隔结束。

9）回到 3)，重复循环。

22.8.2　采样速率

对连续信号采样时，为了使采样值准确地反映连续信号，采样点在时间上必须足够接近，使相邻采样间隔中被采样的信号没有明显波动。或者说，在一个采样间隔内，没有输出变化的信息反馈到控制器中。这意味着，实际采样时必须采用高于连续信号最高频率两倍以上的频率进行采样，这就是香农（Shannon）采样定理（见 4.2.1 节）。在数字控制系统中，采样速率通常比这个高很多。

22.8.3　计算机控制系统

典型的计算机控制系统包括了图 22-15 所示的各个组件，通过键盘输入设置关键值和控制参数。系统软件将提供所需指令的程序，例如，为执行 PID 控制模式的计算机提供显示、识别，并处理操作者输入的指令的程序，提供系统信息，提供启动、关闭指令，提供时钟/日历信息。显示器可能显示以下信息：设定值、实际测量值、采样间隔、误差、控制器设定参数以及校正单元的状态等。显示信息也可能每隔几秒就更新一次。

519

22.9　控制系统性能

控制系统的传递函数受控制器所选择模式的影响，所以系统的响应也受影响（例如，输入是阶跃信号）。考虑图 22-17 的简单系统。

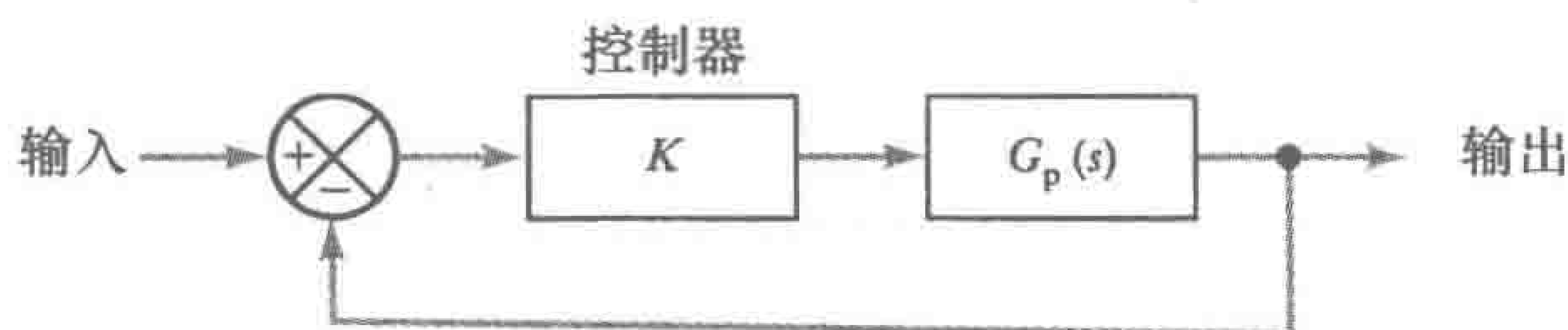


图 22-17　控制系统

在比例控制中，前向通道的传递函数为 $K_PG(s)$，所以反馈系统 $G(s)$ 的传递函数为：

$$G(s) = \frac{K_P G_P(s)}{1 + K_P G_P(s)}$$

假设被控对象有一阶传递函数 $1/(\tau s+1)$，其中 τ 为时间常量（可以表示直流电动机，通常模拟为一阶系统模型，见 20.5.1 节）。具有单位反馈的比例控制，该控制系统的传递函数为：

$$G(s) = \frac{K_P/(\tau s + 1)}{1 + K_P/(\tau s + 1)} = \frac{K_P}{\tau s + 1 + K_P}$$

控制系统为一阶系统。比例控制仅对一阶形式的系统响应产生作用。没有控制器，单位阶跃输入的响应是（见 20.2.1 节）：

$$y = 1 - e^{-t/\tau}$$

现在它为：

$$y = K_P(1 - e^{-t/(\tau/1+K_P)})$$

比例控制用于将时间常量从 τ 减小到 $\tau/(1+K_P)$，较高的 K_P 值使系统响应更快。它也减小了稳态误差。

积分控制中，前向通道的传递函数为 $K_I G_P(s)/s$，系统的传递函数为：

$$G(s) = \frac{K_I G_P(s)}{s + K_I G_P(s)}$$

因此，带 $1/(\tau s+1)$ 传递函数的一阶系统具有比例控制和单位反馈控制，其传递函数为：

520

$$G(s) = \frac{K_I/(\tau s + 1))}{s + K_I/(\tau s + 1)} = \frac{K_I}{s(\tau s + 1) + K_I} = \frac{K_I}{\tau s^2 + s + K_I}$$

现在控制系统是一个二阶系统。对于阶跃输入，系统将产生二阶响应而不是一阶响应。

带微分控制的系统，前向通道传递函数为 $sK_D G(s)$，所以带单位反馈的系统的传递函数为：

$$G(s) = \frac{sK_D G_P(s)}{1 + K_D G_P(s)}$$

对于传递函数为 $1/(\tau s+1)$ 的一阶系统，加上微分控制的系统传递函数为：

$$G(s) = \frac{sK_D/(\tau s + 1)}{1 + sK_D/(\tau s + 1)} = \frac{sK_D}{\tau s + 1 + sK_D}$$

22.10 控制器校正

术语**校正**（tuning）用来描述选择最优控制器参数的过程。比例控制器的系统需要选择 K_P 值；PID 控制器需要选择 K_P、K_I、K_D 三个值。有许多方法可以实现校正。这里仅讨论两种方法：齐格勒法（Ziegler）和尼克尔斯（Nichols）法。他们假设控制系统开环时可以合理地近似为一个带内置时延的一阶系统。基于此假设，他们导出了最优参数。用这种方法设定参数可给出一个 1/4 衰减率的欠阻尼瞬态响应，即第二个超调是第一个超调的 1/4（见 19.5 节）。这种超调标准给出了一个较好的关于短上升时间、短调整时间与合理稳定裕度的折中。

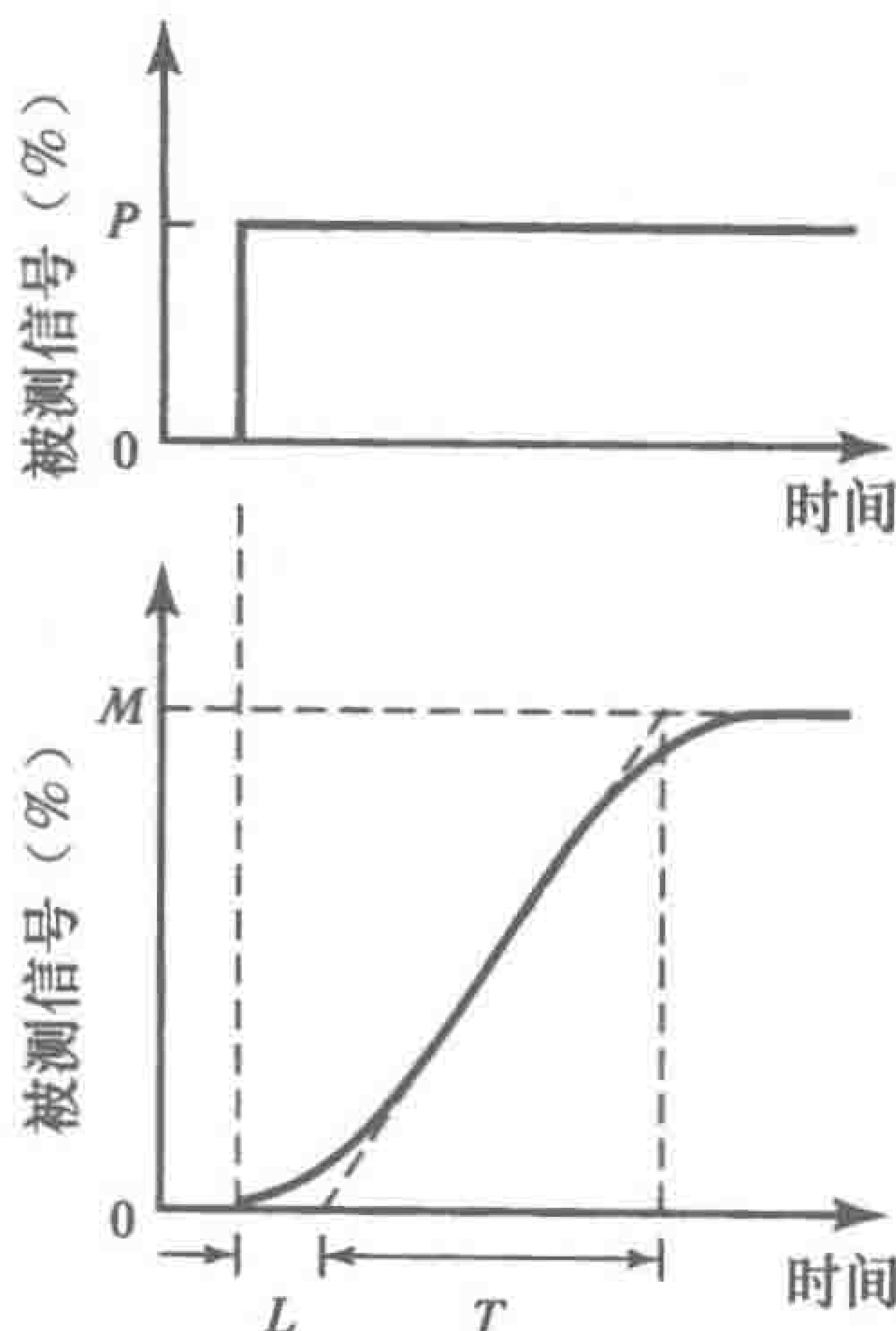


图 22-18　阶跃响应曲线

22.10.1 阶跃响应法

通常在控制器与校正单元之间，控制回路是开环的，所以没有控制动作发生。一个尽可能小的测试信号输入到校正单元，被控变量的响应就确定了。如图 22-18 所示的是测试信号的形式及一种典型响应。测试信号为阶跃信号，阶跃大小为校正单元中的变化百分

比 P。被测变量关于时间的曲线称为**阶跃响应曲线**。被测变量用满量程的百分比表示。

在图 22-18 中最大斜率处做切线，最大斜率 R 等于 M/T。从测试信号输入到切线与时间轴相交点这段时间成为滞后 L，基于 P、R 和 L 值的控制参数设定，表 22-1 给出了齐格勒 - 尼克尔斯建议的标准。

表 22-1 阶跃响应法标准

控制模式	K_P	T_I	T_D
P	P/RL		
PI	$0.9P/RL$	$3.33L$	
PID	$1.2P/RL$	$2L$	$0.5L$

521

例如，当测试信号为控制阀门位置 6% 变化的信号时，其过程响应曲线如图 22-19 所示，确定 PID 三项控制器参数。在图像最大斜率处做切线，得滞后 L 为 150s，斜率 R 为 $5/300 = 0.017/\text{s}$。所以有：

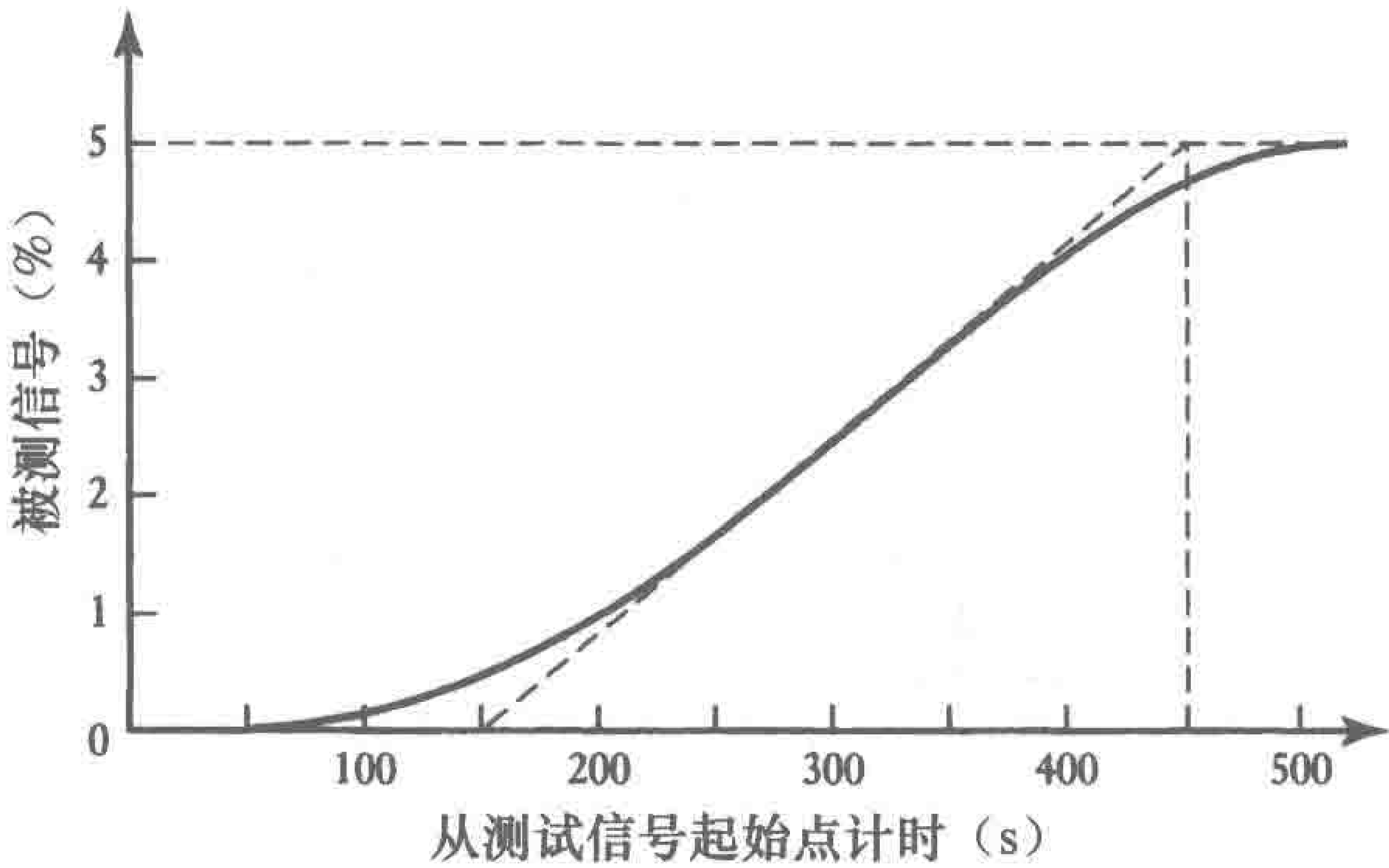


图 22-19 过程曲线实例

$$K_P = \frac{1.2P}{RL} = \frac{1.2 \times 6}{0.017 \times 150} = 2.82$$

$$T_I = 2L = 300\text{s}$$

$$T_D = 0.5L = 0.5 \times 150 = 75\text{s}$$

22.10.2 临界比例度法

采用这种方法，积分和微分参数的作用会先降为最小值。比例常量 K_P 逐渐由设定的低值向高值递增。然而这个过程会引入小干扰，直到出现周期振荡。将比例常量 K_P 设为低值，然后逐步增大。直到出现周期振荡，此时 K_{Pc} 为比例常量临界值，并测得振荡周期时间 T_c。表 22-2 显示了齐格勒 - 尼克尔斯标准建议的控制器参数与 K_{Pc} 的关系。临界比例带为 $100/K_{Pc}$。

表 22-2 临界比例度法标准

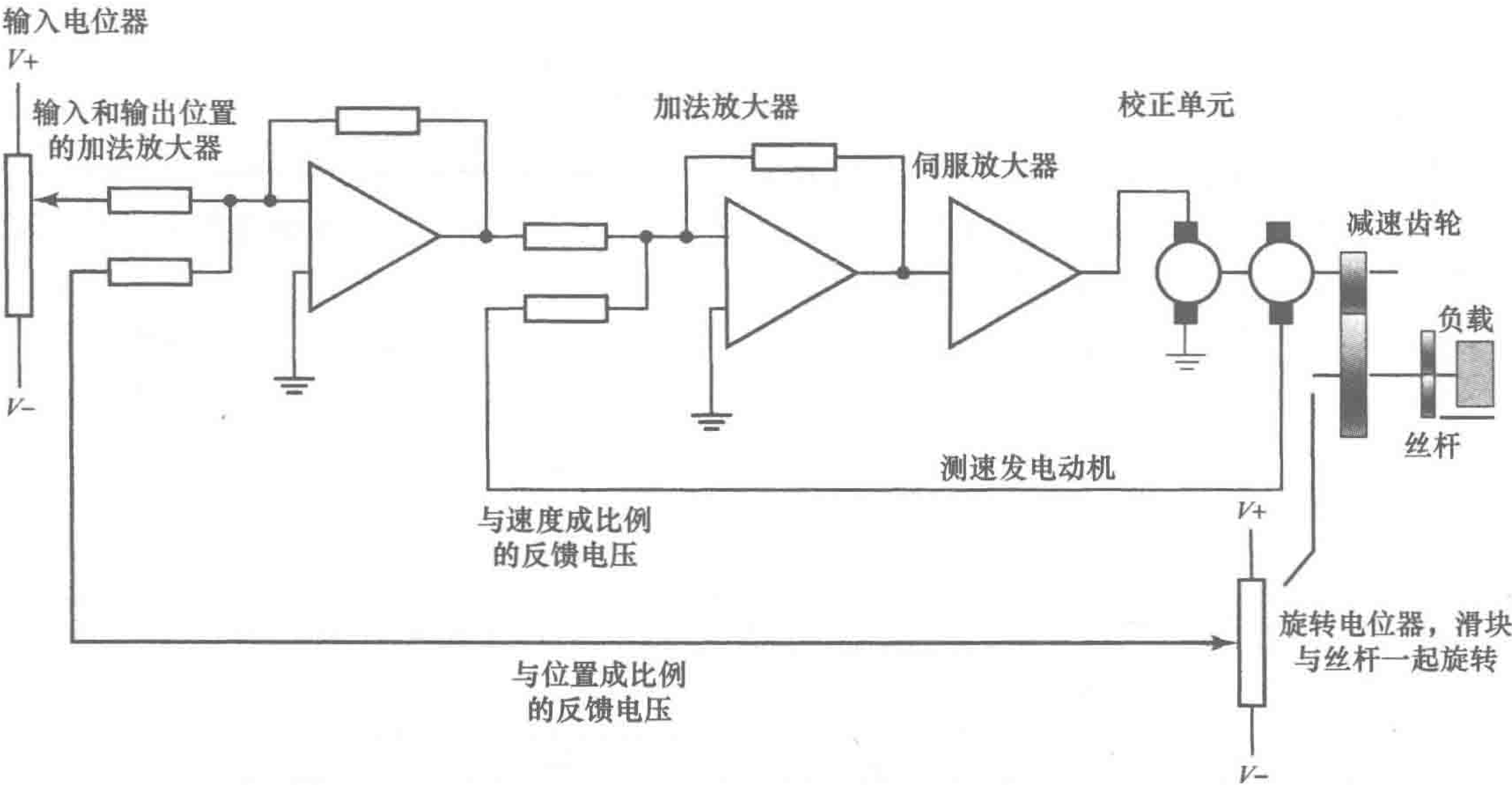
控制模式	K_P	T_I	T_D
P	$0.5K_{Pc}$		
PI	$0.45K_{Pc}$	$T_c/1.2$	
PID	$0.6K_{Pc}$	$T_c/2.0$	$T_c/8$

例如，用临界比例度法校正三项控制系统，当 $K_{Pc} = 3.33$ 时出现振荡，振荡周期为 500s。控制器的合适参数是多少？用表 22-2 中的标准，可得 $K_P = 0.6K_{Pc} = 0.6 \times 3.33 = 2.0$，$T_I = T_c/2.0 = 500/2 = 250\text{s}$，$T_D = T_c/8 = 500/8 = 62.5\text{s}$。

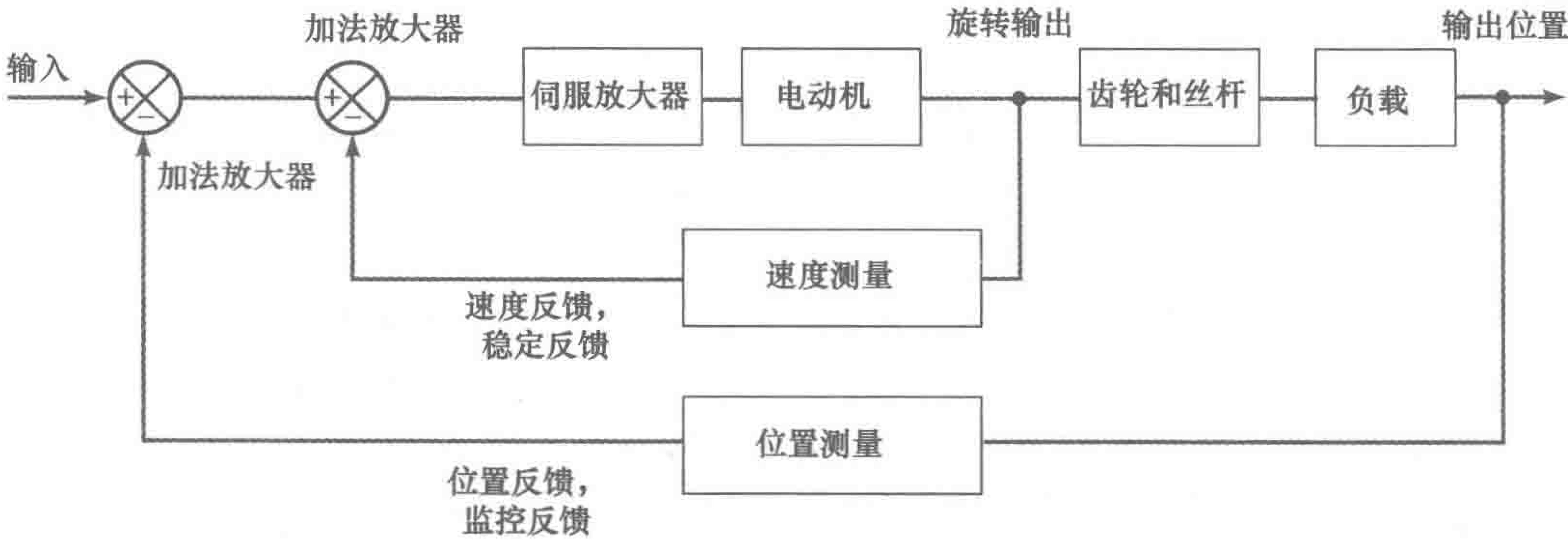
22.11 速度控制

考虑如何用电动机控制负载移动这样一个问题，用电动机控制负载移动。因为电动机系统可能是二阶的，所以比例控制将会使系统输出滞后一定时间才能达到期望的位移，对于一个阶跃输入，系统先振荡一段时间才能达到期望的目标。系统需要一段时间来响应输入信号。用比例微分控制而不是比例控制，可以得到更快速度的响应，而且振荡还更小。但是，通过对位移

的变化率进行测量的第二个反馈回路，也能达到同样的效果。这就是所谓的**速度反馈**。图 22-20 给出了这样的一个系统。速度反馈采用测速发电机产生一个与电动机转速成比例的信号，而位移量可以用旋转电位器进行测量。



a）系统说明图



b）系统框图

图 22-20　速度反馈系统

22.12　自适应控制

有很多控制情况中被控对象的参数随时间或负载等发生改变，例如搬运负载的机器人的机械臂，其负载是变化的。如果被控对象的传递函数发生变化，则系统需要重新校正，以便得到优化的比例、微分和积分常量。至今所讨论的控制系统，在操作者决定调整三项常量之前，被校正的系统将保持原参数。另一种方案是**自适应控制系统**，能够自适应改变参数以适应环境。

自适应控制系统是基于微处理器的控制器，这类设备可以使控制模式和控制参数适应环境，跟随环境变化而不断修正自身的参数。

自适应控制系统具有三个操作步骤：

1）用一个假想条件初始化控制器。

2）将实际系统性能与期望性能进行实时比较。

3）为了减少期望与实际系统性能的差异，需要自动调整控制系统模型与参数。

例如，工作在比例模式的控制系统，比例常量 K_P 可以自动调整以适应环境。自适应系统可采用多种形式，下面是常用的方法：

1）变增益控制。

2）自校正。

3）模型参考自适应系统。

522 ~ 524

22.12.1 变增益控制

变增益控制也称为**预编程自适应控制**，通过对某些过程变量的辅助测量，预先设定改变控制器的参数。图 22-21 表示了这种方法。唯一被调整的原始参数是增益，即比例常量 K_P，所以称为变增益控制。

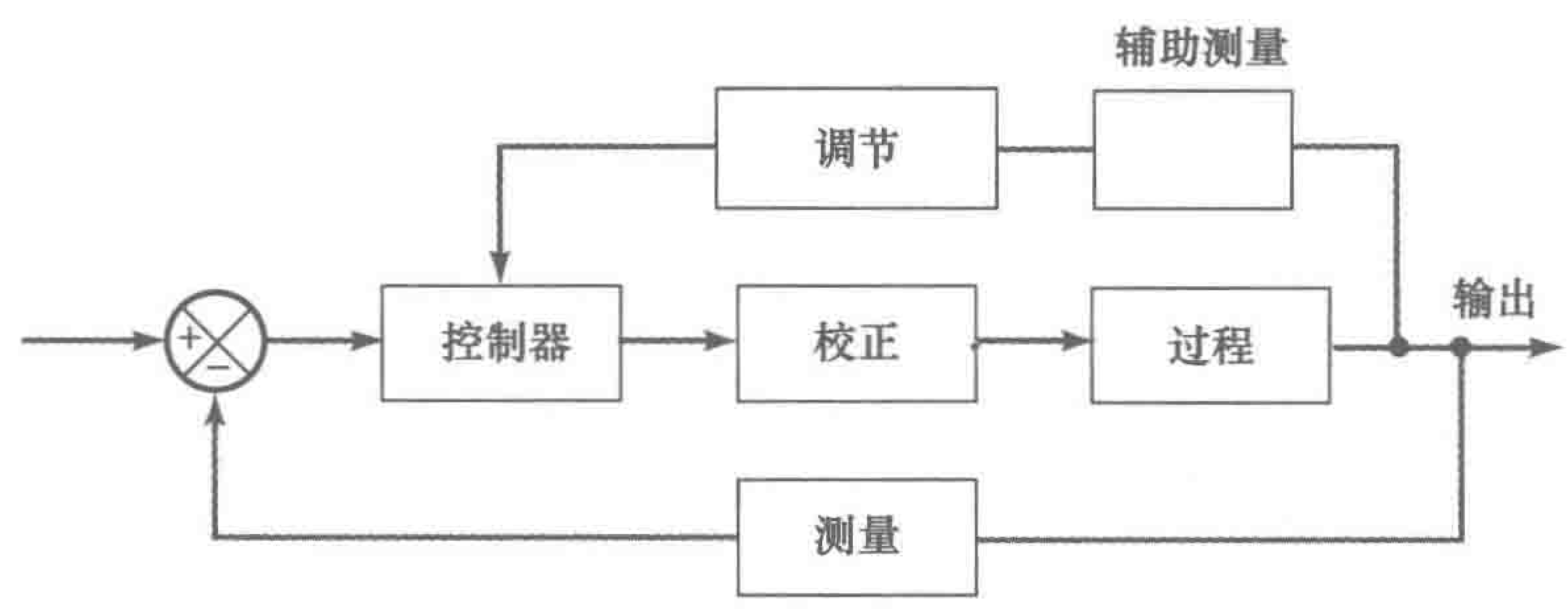


图 22-21 变增益控制

例如，对于控制负载定位的控制系统，系统参数可根据不同的负载值和一系列存入控制器存储器的数据表格计算得出。称重传感器测量实际负载，并向控制器发出信号指示一个质量值，控制器根据该值选择合适的参数。

这个系统的缺点在于，必须确定很多不同的工作条件的控制参数，使控制器可选择一个参数以适应主要条件。该系统也有一个明显优点，当条件发生变化时，系统参数能够快速地修改。

22.12.2 自校正

在**自校正控制**系统中，系统根据监测的控制变量和控制器的输出连续地校正它们的参数。图 22-22说明了该系统的特征。

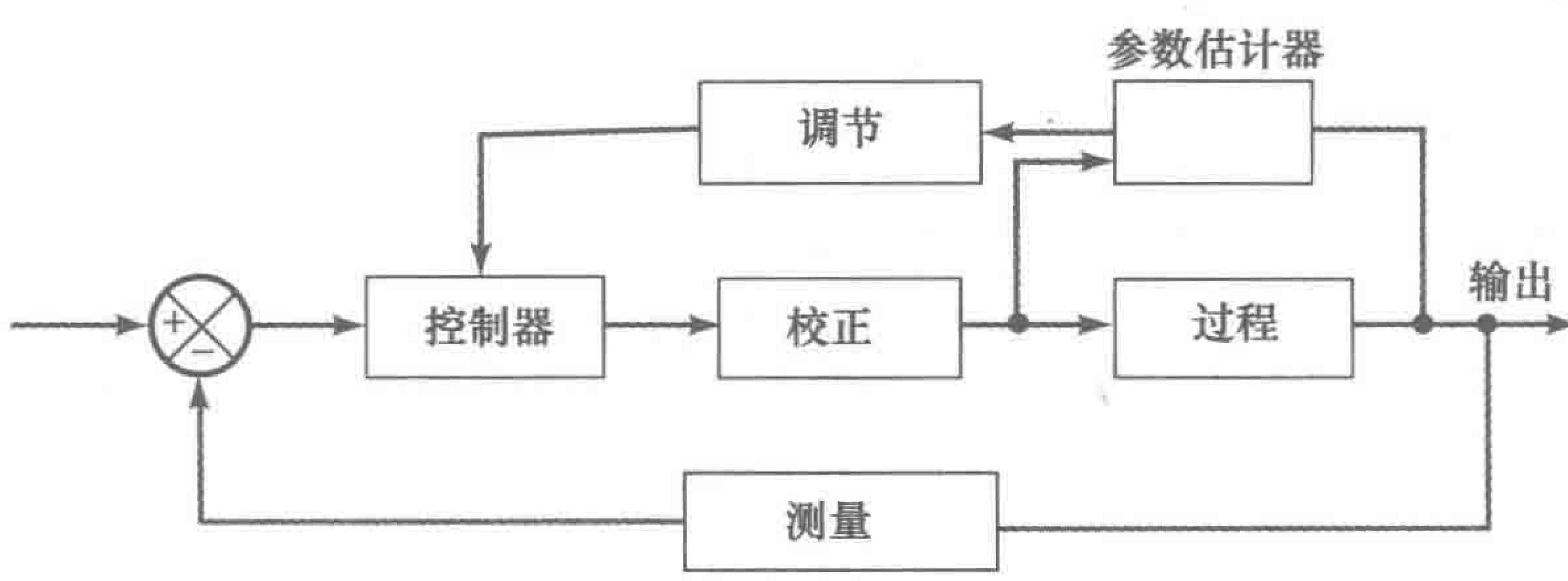


图 22-22 自校正

自校正常用于商用的 PID 控制器中，通常也称为**自动整定**。当操作者按下按钮时，控制器向系统发出小的扰动信号，并测量系统响应。将实际响应与期望响应比较，采用改进的齐格勒－

525 尼科尔斯规则调整控制参数，使实际响应接近于期望响应。

22.12.3　模型参考自适应系统

在**模型参考自适应系统**中，设计了一个十分精确的系统模型。设定值同时输入实际系统和模型系统中，实际输出与模型输出进行比较，差值用于调整控制器参数并缩小误差。图 22-23 说明了这个系统的特征。

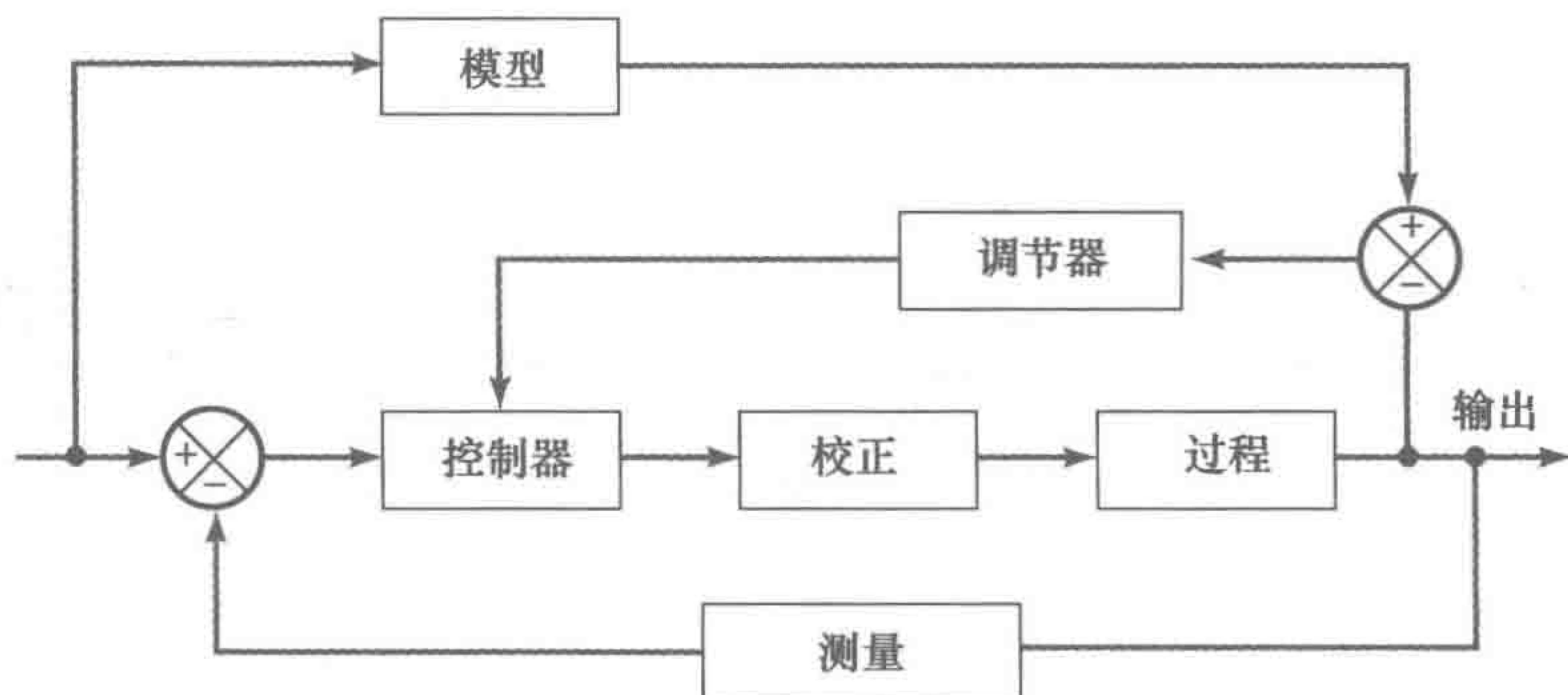


图 22-23　模型参考控制

本章小结

稳态误差是指期望设定值输入与稳态输出的差值。

控制模式包括：1）**两步模式**，控制器仅提供开关校正信号；2）**比例模式**（P），校正信号与误差成比例关系；3）**微分模式**（D），校正信号与误差变化率成比例关系；（4）**积分模式**（I），校正信号与误差的时间积分成比例关系。PID 控制系统的传递函数为：

$$\text{传递函数} = K_P + \frac{1}{s}K_I + sK_D = K_P\left(1 + \frac{1}{T_I s} + T_D s\right)$$

数字控制器主要通过对测量值进行采样，与给定值比较得到误差，基于该误差值和存储的前一个输入、输出值进行计算来获得输出信号，然后等待下一个采样周期。

校正（tuning）用于描述选择最佳控制器参数 K_P、K_I、K_D 的过程。

自适应控制用于系统自适应地改变参数以适应主要的环境。三种常用的形式是变增益控526制、自校正控制和模型参考自适应系统。

习题

22.1　两步控制（开/关控制）的局限性是什么？这种系统通常用于什么情况？

22.2　一个两步模型控制器当温度降到 20℃时打开房间加热器，当温度达到 24℃时关闭加热器。当加热器打开时，房间空气温度以每分钟 0.5℃的速度增加，但加热器关闭时，以每分钟 0.2℃的速度下降。如果控制系统的时滞可以忽略，那么：（a）加热器由打开到关闭过程一共经历多长时间？（b）加热器由关闭到打开经历多长时间？

22.3　一个两步模型控制器通过打开/关闭阀门控制水箱水位。阀门打开时水以 $0.4\text{m}^3/\text{s}$ 的速度流入水箱。水箱横截面积为 12m^2，水以 $0.2\text{m}^3/\text{s}$ 的恒速流出。当水位下降到 4.0m 时阀门打开，上升到 4.4m 时阀门关闭。从阀门打开到关闭需要经历多长时间，从关闭到打开需要经历多长时间？

22.4　一个比例控制器用于控制水箱水位，水位从 0 ~ 4m。要求水位为 3.5m，控制器在水位达到 3.9m 时完全关闭阀门，当水位下降到 3.1 米时完全打开阀门。问将需要什么样的

系统传递函数？

22.5　比较并描述（a）比例控制；（b）比例积分控制；（c）比例积分微分控制三者的特点。

22.6　已知三项控制器的过程响应曲线的时滞 L 为 200s，斜率 R 为 0.010%/s，当信号为 5% 控制阀门的位置变化时，确定系统 K_P、T_I、T_D 值。

22.7　使用临界比例度法调节三项控制系统，当比例临界值为 5 时出现振荡，振荡周期为 200s。求合适的 K_P、T_I、T_D 值？

22.8　解释以下形式的自适应控制系统的基本原理：（a）变增益；（b）自校正；（c）模型参考。

22.9　一阶系统直流电动机的传递函数与输出位置相关。在 $1/s(s+s\tau)$ 的输入信号作用下转动负载。如果时间常量 τ 等于 1 秒，电动机是有单位反馈和比例控制器的闭环控制系统，确定使闭环响应有 25% 超调的比例常量值。

22.10　一个小型超声波电动机用于照相机（见 24.2.3 节）自动对焦时移动镜头，驱动一个惯量十分小的环，传递函数与角度位置有关，输入信号可以用 $1/cs$ 表示，其中 c 为与摩擦转矩和角速度相关的比例常量。假设电动机采用带单位反馈的闭环系统来控制，如果用比例控制，请问能产生什么类型的响应？

527

第23章 人工智能

本章目的

通过本章的学习，读者将能够：

- 了解什么是智能机，智能机能够做什么。
- 了解神经网络的含义及其与模式识别之间的关联。
- 了解模糊逻辑。

23.1 什么是人工智能

智能机是由什么组成的？字典把“具备了推理的能力”定义为智能。我们认为一个人的智力愈高，他就具备着愈强的学习和从已有知识推广到未知领域的能力，而且有更强的推理能力，可以通过对错误的学习，对事情发生的可能性进行预测。我们可以把类似的准则应用在智能机上：所谓**智能机**，就是一种具备了推理能力的机器。

一个集中供热系统可以对自己的行为进行决策。例如，锅炉从恒温器那里获得信息，从而选择“加热”或者“停止”。然而，它却算不上是智能机，因为它不能在众多的情况下进行分析推理并做出决策。比如说，它并不能识别出来自恒温器的输入模式，也就自然不能预测出未来是否打开或关闭该加热器。它只是做了别人让它做的事情，它自己并不会思考。

在这一章中，我们简要了解与智能机相关的一些基本概念。

23.1.1 自我调控

在之前的章节中我们讨论的闭环反馈系统就是自我调控系统，因为它们可以自动调节输出，并使之达到一个需要的数值。集中供热系统能够使室内温度保持在一个预先设定的数值。然而，这类系统仍然不能称为智能系统，它们也仅仅是做了别人要求它们完成的事情。

23.2 感知和识别

所谓智能系统的**感知**，就是通过传感器获取信息、将已有的信息组织起来，并做出相应的决策。例如，在生产线的控制系统中，可能使用摄像机来观察传送带上的零件，观察到的信号触发系统对零件进行相关计算，从而识别一些特征，而这其中可能就包含了零件的关键信息。然后，通过比较、分析这些信息，控制系统可以判断这些零件组装的是否正确，或者判断它们分别是什么零件。接着，控制系统就可以做出决策来丢弃那些错误的零件或者将不同的零件分拣到不同的盒子里。

528

因此，机电一体化系统的感知包括对系统及其所处环境内传感器有效信息的获取、信息解码和有用信息的处理。这些有用信息可用在系统的其他地方，使系统能做出正确的决策。

23.2.1 识别

一旦智能机收集和整理了信息，就可以根据这些信息做出相应的决策。这个过程就称为**识别**。对于感知和识别来说，最重要的就是**模式识别**。那么，采集数据的模式是什么呢？

人类非常善于模式识别。想象一位保安正目不转睛地盯着电视监控器，他能够从监视器里识别出不同寻常的“模式”，比如一个不速之客出现在了本该没有人的区域，或者一件物品被无端移动了等。而我们也需要智能机能够做到这些。例如飞机上的自动飞行系统，它可以监控大量的信息，并基于从信息中提取的“模式”来判断如何调整飞机的飞行。

机器的模式识别是怎样完成的呢？机器内部存储系统保存了一套已知的模式，同时机器还可以将接收到的模式与之相对比，根据匹配程度来识别模式。内存中的模式可以通过建模或者训练获得。在训练过程中，通过对对象及环境数据的获取，给出识别的编码。例如，为了识别出硬币，智能机需要收集硬币的直径和颜色等信息。因此如果你收集到了“直径2.25cm”和“表面一定程度上呈红色”这两项主要信息，那么就可以确定这是一枚铜质的1英镑的硬币。当然，仅仅做到这些是不够的，一台智能机将需要识别出那些污损的硬币，如结合上述信息仍然可识别出1英镑的硬币来。

23.2.2 神经网络

在前面识别硬币的例子中，我们只考虑了直径和颜色这两个方面，而在一些更为复杂的情况下，我们需要考虑更多的方面。人类的大脑经常要对多维信息进行分类或者排序。这个过程是通过**神经网络**实现的。而人工神经网络已经应用在了智能机领域中。这种网络不需要编程，但是却可以从例子或者训练中进行学习和总结。神经网络（如图23-1所示）是由大量的相互连接的处理单元构成的，一些单元的输出作为另一些单元的输入。网络中的每个处理单元在输入端接收信息，并与加权因子相乘。如果使用了AND操作，系统就会求出各个输入的加权和，然后根据结果是否超过某个值或者被判定为正值来确定是否为输出端置1。例如，一个输入为1.0，加权因子-1.5，得到-1.5；另一个输入为1.0，加权因子为1.0，得到1.0；第三个输入为1.0，加权因子为1.0，得到1.0，那么这三条输入的加权和就为-1.5+1.0+1.0=0.5，根据前面的规则（结果为正值，则输出端置1），则输出为1；如果输入为1×(-1.5)、0×1.0和0×1.0，那么加权和就为-1.5，相应的输出端为0。该网络可以从样本进行学习编程，因此具备学习能力。

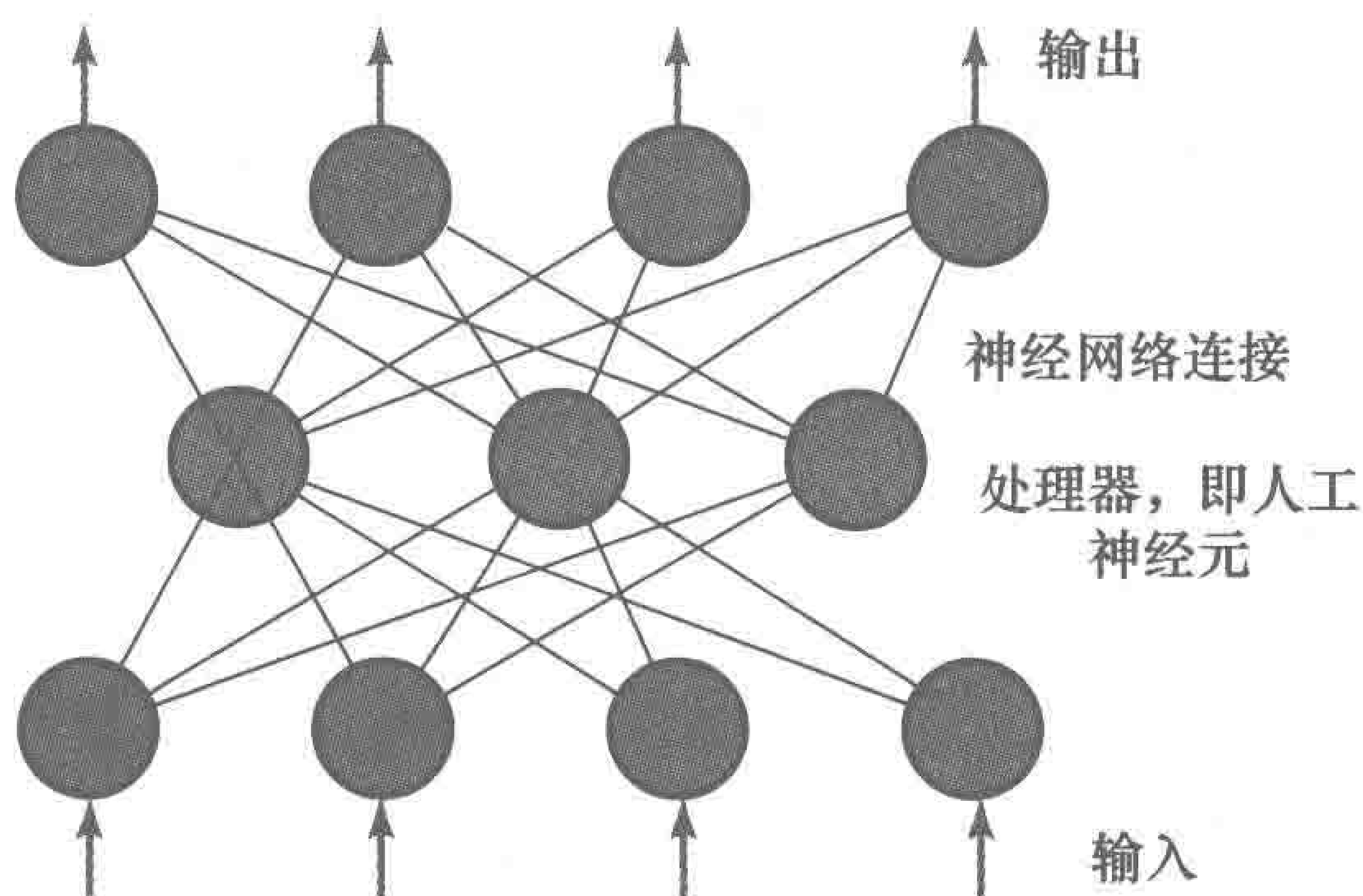


图23-1 神经网络

23.3 推理

推理是一种从已知求取未知的过程，推理机也分为很多种。

23.3.1 推理机理

确定性推理的一个典型例子是使用“if-then”规则。相应地，我们可以说“if 硬币直径为2.25cm，then 它是1英镑的硬币”。命题的前一部分若为真；则后一部分为真；前一部分为假，则后一部分为假。在这种推理模式下，我们得到的 true-false 情况，并假定“没有默认的已有知识”，因此当做出推论时不会出现例外情况。因此，如果出现一枚来自其他国家的硬币，而这枚硬币的直径也恰好相同，那么我们就没办法推理了。

非确定性推理允许我们基于概率进行预测。如果你抛出一枚硬币，它落地的时候有字的一面要么朝上要么朝下，因此字面朝上（或朝下）的概率就是1/2。为了得到这个数字，我们可以重复地抛硬币很多次，这其中会有一半的次数是字面朝上的。图23-2a将这一情形用概率树表达出来。如果我们抛的是一个骰子，那么当它落地时表示“6”的面朝上的概率就是1/6。

529~530

图 23-2b 是抛骰子的概率树。树的每一分支都标有概率值，各分支之和为 1。

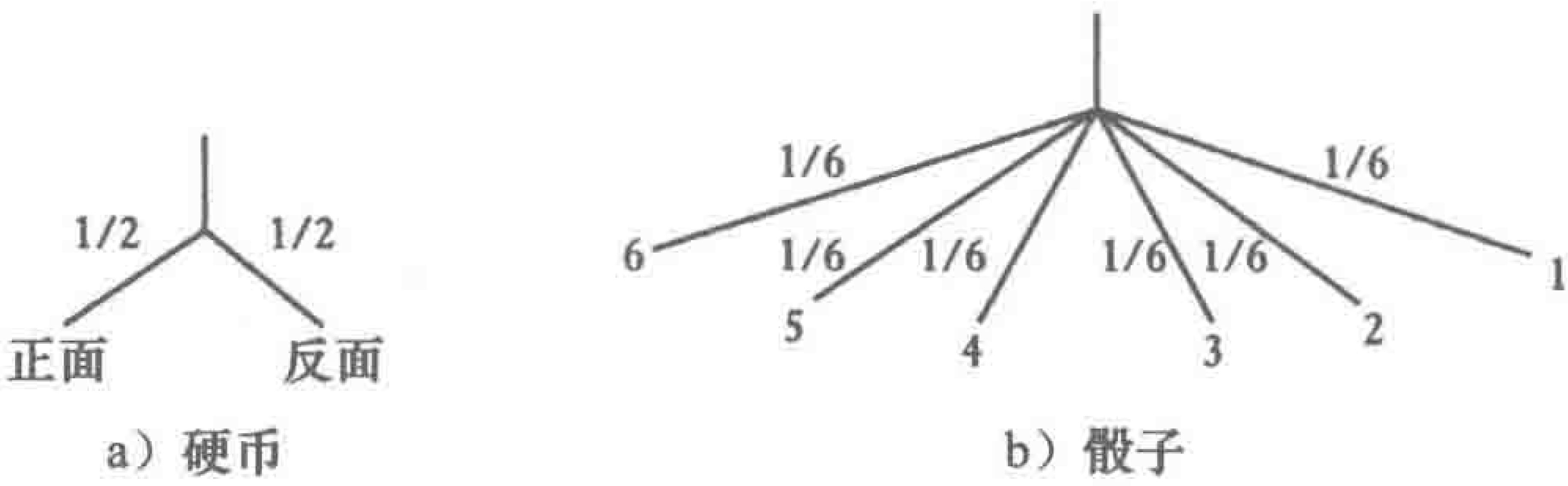


图 23-2 概率树

在前面关于硬币的例子中，我们可能会考虑直径为 2.25cm 的硬币是 1 英镑的概率为 0.9。在机电一体化系统中，我们可能对环境监视 1000h，其中 3h 环境温度过高，那么温度高的概率就为 0.003。

有时我们可能知道一件事情发生的概率，然后想知道由此发生另一件事情的概率。比如，在机电控制系统中，我们需要求取当传感器检测到过热时系统缘于低压的概率，而我们需要知道，还有其他原因会导致过热。我们通过图 23-3 来表达这个问题。

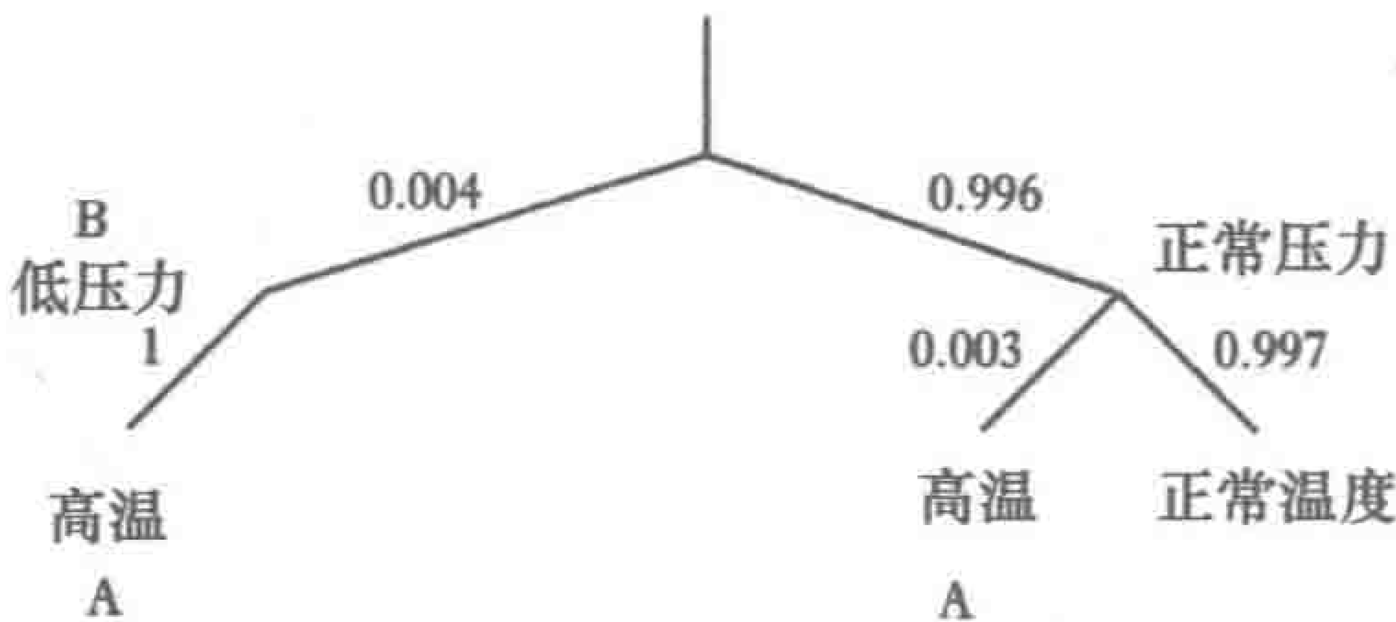


图 23-3 条件概率树

贝叶斯定理可用来解决这个问题。这个定理可以表述如下：

$$p(B \mid A) = \frac{p(A \mid B) \times p(B)}{p(A)}$$

其中 $p(A \mid B)$ 是当 B 发生时，A 发生的概率；$p(B \mid A)$是当A 发生时，B 发生的概率；$p(A)$ 是A 的发生概率；$p(B)$ 是 B 的发生概率。因此如果高温的发生概率 $p(A)$ 为 $1-0.997\times0.996=0.006\,988$，即 1000 次中有 6.988 次发生高温；低压的发生概率$p(B)$ 为0.004，即 1000 次中有 4 次发生低压。因为我们可以确定，低压必然引起过热，即 $p(A \mid B)$ 为 1，所以可推算出当检测到过热时，必然有 $p(B \mid A)=(1\times0.004)/0.006\,988=0.572\,41$，系统会发生过热缘于低压的条件概率为 0.572 41。

23.3.2 基于规则的推理

531

基于规则的系统的核心是规则的集合。当这些规则与系统的客观状态相结合时，就可以得到所需的控制输出，例如驱动器或者控制输出量。图 23-4 举例说明了这一顺序。事实和规则数据库的结合称为智能机的知识库。将输入的事实和规则结合起来，做出决策，并将其输送给驱动器的过程，就是推理的过程。

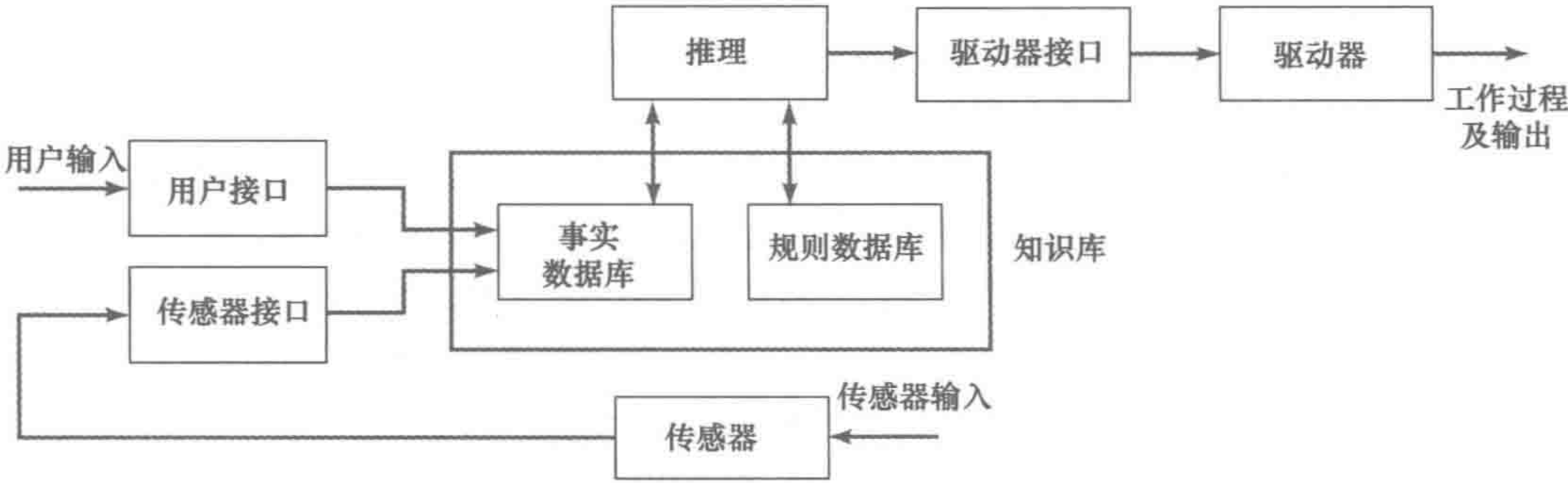


图 23-4 基于规则的系统

规则通常存储为“if-then”语句形式。例如，下面是集中供热系统的一系列规则：

```
if 锅炉工作
then 水泵就要打开

if 水泵工作并且室温低于 20℃
then 打开阀门

if 锅炉不工作
then 水泵也不工作

等等
```

事实数据库可能包含下面的事实：

```
室温低于 20℃
定时器工作
阀门打开
锅炉工作
水泵工作
```

规则同样可以用涉及概率或者模糊逻辑的命题形式。

Lotfi Zadeh 于 1965 年提出了一种推理形式，它已成为众所周知的**模糊逻辑**。模糊逻辑的主要思想就是在命题中不再需要明确地区分出 true 或 false，而是其真、假均被赋予一定的程度，这样就可以在一定程度上衡量介于真、假之间的值。**隶属函数**定义一个值，用于判断其是否属于某个集合。我们可以定义一个 0～20℃ 的温度集合，还可以定义一个 20～40℃ 的温度集合。那么对于 18℃ 来说，对第一个集合的隶属度就是 1，对第二个集合的隶属度为 0。然而，用模糊逻辑我们可以定义交叠集合：冷（0～20℃），暖（10～30℃），热(20～40℃)。那么 18℃ 就可以在两个集合中找到。如果定义的模糊集合的隶属函数如图 23-5 所示，则 18℃ 在冷中的隶属度为 0.2，在暖中为 0.8，在热中为 0。基于此，我们可以设计出能够激发更合适动作的规则。例如，冷的隶属度为 0.2 时可以设置成低功率加热，0.6 时则设置为大功率加热。 532

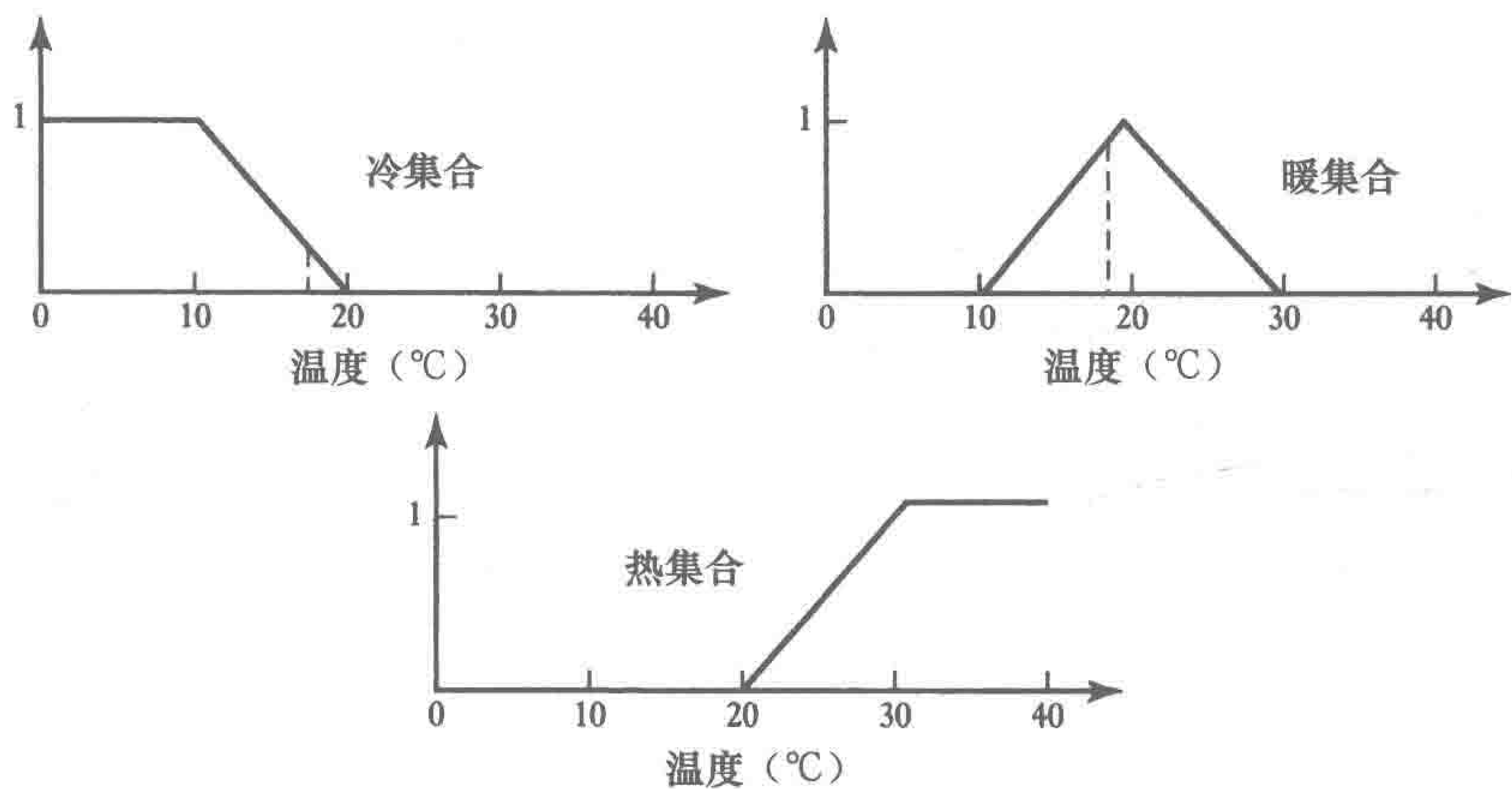


图 23-5 模糊集合隶属函数

模糊逻辑在很多产品中已经得到了广泛应用。例如，家用洗衣机可以检测出衣物材质类型、脏的程度和负载大小，从而合理地调整洗衣工作周期的长短。

23.4 学习

能够学习并扩充自身知识库的机器要比不能学习的机器有优势。**学习**可以理解为根据经验

来适应环境的过程。机器可以通过多种方式来进行学习。

一种比较简单的学习方法是将新的数据输入存储器并积累起来。机器还可以通过外部输入数据对内部参数进行调整的方式来学习。

当根据概率来进行推理时，机器可以用另一种学习方式，它们可以根据不断发生的事实来更新接下来某件事发生的概率。举一个简单的例子。我们有一个袋子，里面有 10 个球，一个是黑色，其余都是红色。我们第一次进去抓球，抓到黑色的概率是 1/10，如果我们发现抓到的是红色的，那么继续抓的时候，抓到黑色的概率就是 1/9。机器可以根据前面几次抓球的情况，实时更新概率值。23. 3. 1 节说到的贝叶斯定理可以用来更新机器的数据，此时可以这样表达：

$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E)}$$

其中 H 是前提假设，E 是当前的实例。其中 $p(H|E)$ 是当 E 发生时，H 为真的概率；$p(E|H)$ 是假设 H 为真的时，实例 E 发生的概率；$p(H)$ 是假设 H 为真的概率；$p(E)$ 是实例 E 发生的概率。这样，机器就可以在每次得到新信息时更新假设 H 为真的概率。

机器的另一种学习方法就是通过样本实例来学习。通过带有实例的训练，可以获得训练结果并将其提供给机器，使机器建立起更完善的规则或者作为已经遇到的事件的结果。模式识别通常就是使用这种学习方法。比如说，通过给出有像素点矩阵组成的数字“2”，机器就可以学习识别出数字“2”。神经网络也涉及这种实例学习的方式（见 23. 2. 2 节）。

533

此外，机器也可以通过将已解决的问题和新问题进行类比的方式来学习。

本章小结

智能机是具备推理能力的机器。智能系统的**感知**，是通过传感器收集信息，并对信息进行组织，然后做出决策。**推理**是从已知探求未知的过程。应用“if-then”规则是**确定性推理**的一个例子。**非确定性推理**则允许我们根据概率做出预测。**模糊逻辑**就是在命题中不再需要明确地区分出真或假，而是其真、假均被赋予一定的程度，这样就可以在一定程度上衡量介于真、假之间的值。最后，**学习**可以理解为根据经验来适应环境的一种过程。

习题

23. 1　考察你所在国家的硬币，并制作一个模式识别表。

23. 2　求概率：（a）掷一个六面骰子得到一个“6”的概率；（b）掷两个六面骰子得到一个“6”的概率；（c）从一个装有 9 个红球 1 个黑球的袋子里拿出一个黑球的概率。

23. 3　如果机电一体化系统显示高温的概率是 0. 01，那么不显示高温的概率是多少？

23. 4　某机器在被监测的 2000 小时内冷却系统有 4 小时显示有泄漏发生，求泄漏发生的概率。

23. 5　冷却系统的泄露概率是 0. 005，系统显示高温的概率是 0. 008，如果泄露必然引起高温，那么高温是由泄露引起的概率是多大？

23. 6　由 A、B 和 C 组成的机器发生故障的概率为 0. 46，如果 A 起作用的概率是 0. 5，而且 A 发生故障的概率是 0. 7，那么 A 要为某次故障负责的概率是多少？

23. 7　为一套温控系统提出 if-then 规则。这套系统用于操控锅炉和阀门，且仅当系统工作在某个时间段时，阀门允许水循环流经中央散热器。

534

第六部分　总　　结

第24章　机电一体化系统

本章目的

通过本章的学习，读者将能够：

- 从机电一体化的观点出发，提出设计问题的可能解决方案。
- 分析机电解决方案的案例研究。

24.1　机电设计

本章汇集了许多书中讨论的涉及设计问题的机电解决方案的主题，并给出了案例研究。

24.1.1　定时开关

设计一种设备，使这种设备能够使驱动器（比如电动机）开启，并维持一段指定时间，可能的方案有：

1）旋转凸轮。

2）可编程逻辑控制器（PLC）系统。

3）微处理器。

4）微控制器。

5）定时器，比如555定时器。

机械方案中可以采用旋转凸轮（见图24-1）。凸轮以一定的速度旋转，凸轮从动件用做驱动一个开关，开关闭合的时间则依赖于凸轮的形状。过去这种方法应用十分广泛。

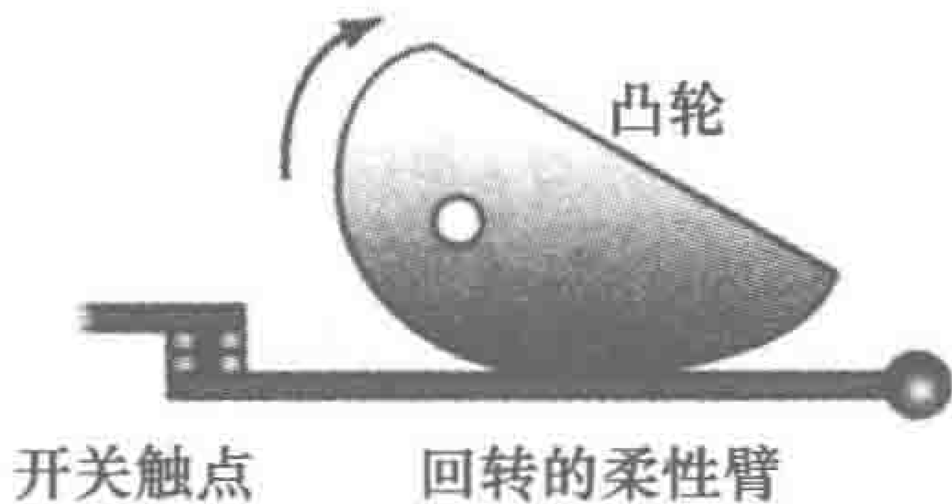


图24-1　凸轮控制的开关

PLC系统可包含如图24-2所示的电路及给定的梯形图程序。PLC系统是优于旋转凸轮的，PLC可以单纯地在程序中改变定时器的设定值即可实现，而机械方案中若要改变开关时间则需一个不同的凸轮来实现。软件方法比硬件方法更容易实现。

基于微处理器的方案则应包含一个有存储芯片和I/O接口的微处理器。程序用于开启某个输出量，然后经过一段时间的延迟后再关闭，而延迟时间是由一段程序中的定时循环产生的。它通过一定时间循环的跳转次数来产生要求的延时。因此，用汇编语言我们可以得到：


```
DELAY    LDX     #F424 ;F424 为循环次数
LOOP     DEX
         BNE     LOOP
         RTS
```

DEX将变址寄存器减1，若不等于零则跳转，每个命令都占用4个时钟周期。因此，一次

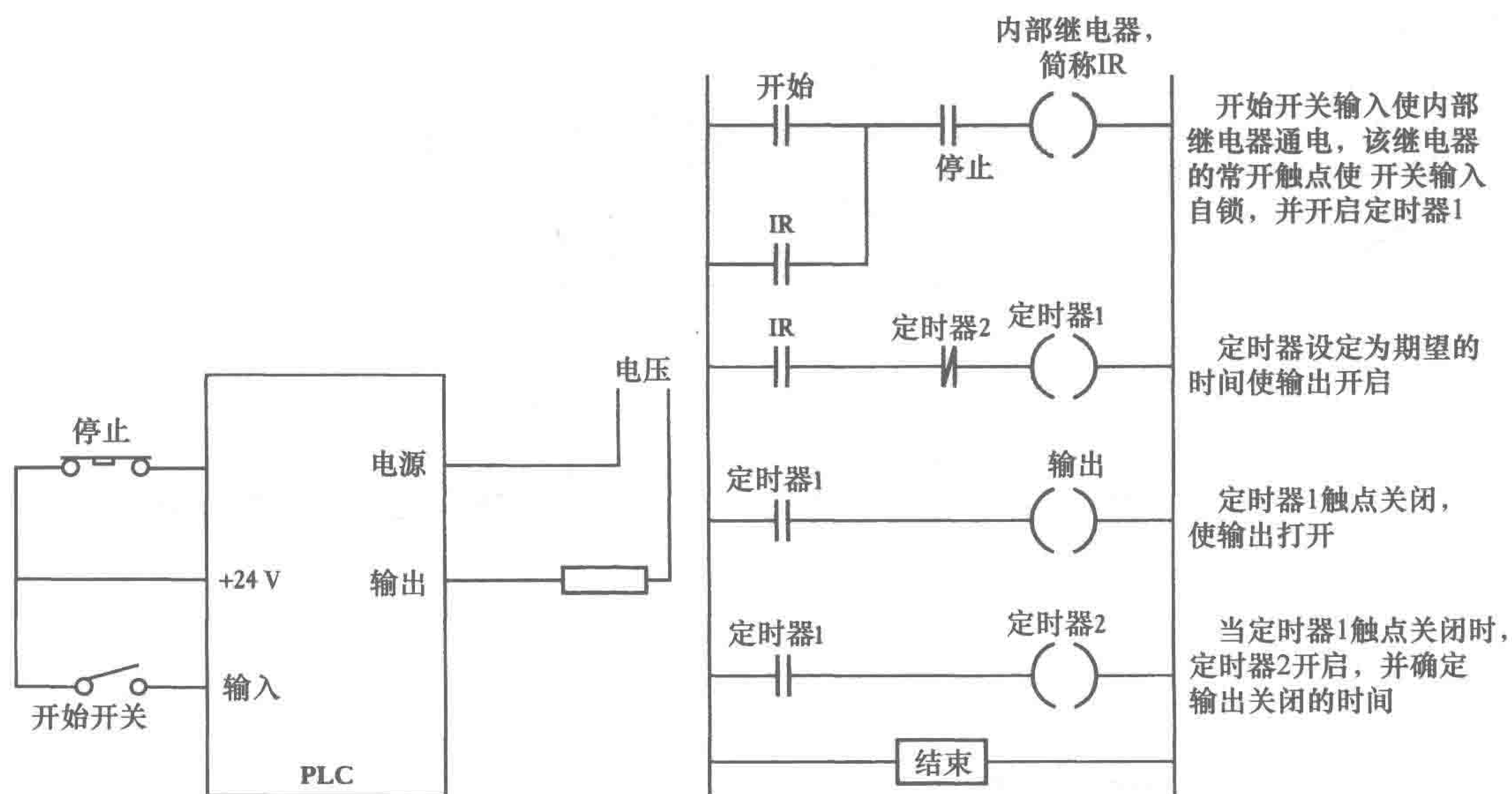
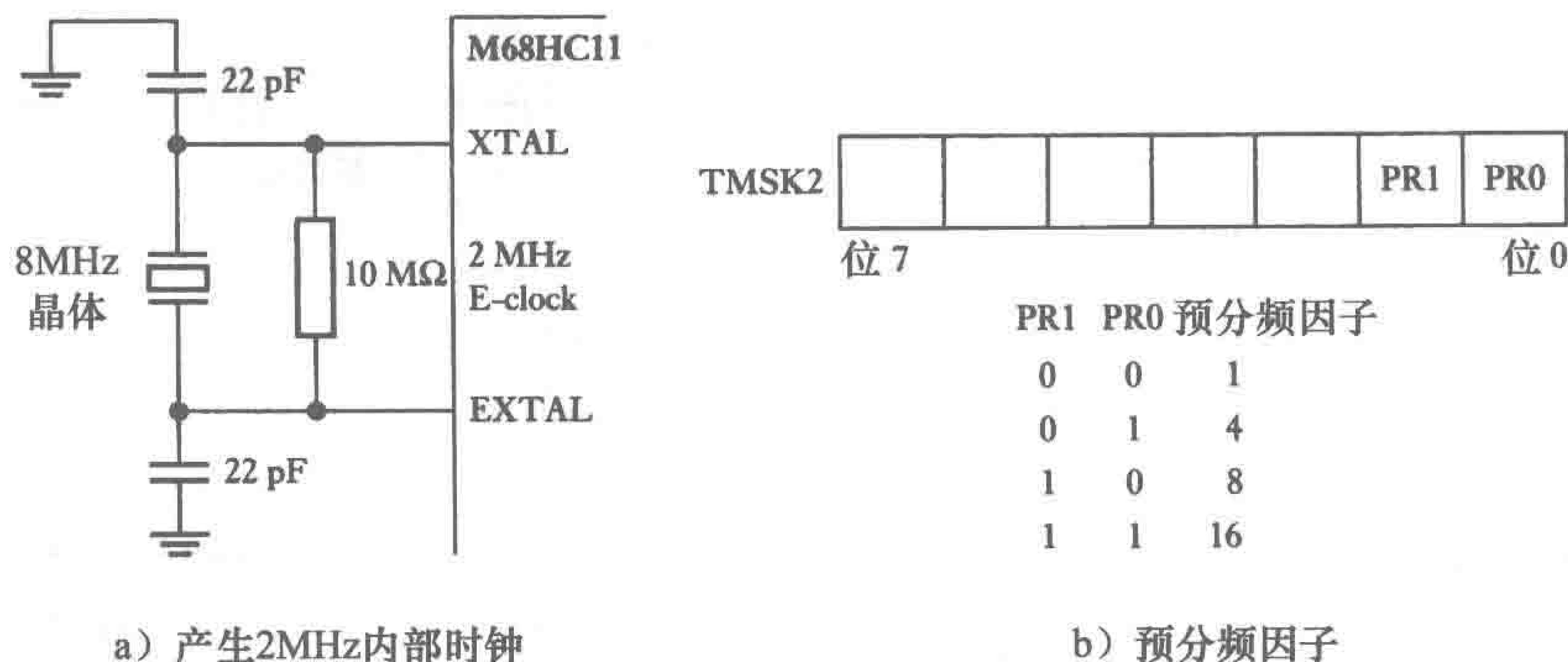


图 24-2　PLC 定时器系统

循环会占用 8 个时钟周期，直到 n 个循环后 $8n+3+5$ 等于 F424（其中 LDX 占用 3 个周期，RTS 占用 5 个周期）。在 C 语言中，我们可以用 while 函数来写这种循环程序。

另一种方法是应用微处理器（如 MC68HC11）中的定时系统。这种方法是以系统的 E 时钟信号进行操作的 16 位 TCNT 计数器为基础的（见图 24-3a）。可以通过设置定时器中断屏蔽寄存器 2（TMSK2，地址为 \$1024）的 PR0、PR1 位对系统 E 时钟预分频（见图 24-3b）。当处理器复位时，TCNT 寄存器从地址 \$0000 开始计数，并且一直计数到计数器的最大值 \$FFFF。当再来一个脉冲时产生溢出，定时器又从 \$0000 开始读。当定时器溢出时，定时器溢出标志 TOF 置为 1（定时器中断标志寄存器 TFLG2 的第 7 位，地址为 \$1025）。因此，当预分频因子为 1，E 时钟频率为 2MHz 时，32.768ms 后产生溢出。



a）产生2MHz内部时钟　　b）预分频因子

图 24-3　晶振与预分频

利用这种定时器进行定时的一种方法就是要对 TOF 标志进行轮询。当 TOF 置位时，程序将计数器的值增加。程序中通过将 1 写入 TFLG2 寄存器第 7 位将标志 TOF 复位。因此，定时操作是由等待溢出标志置位所需数量的程序组成的。

一种更好的定时方法是使用输出比较功能。微控制器的端口 A 可被用作通用输入/输

出或用作定时。定时器有输出引脚 OC1、OC2、OC3、OC4 和 OC5，内部寄存器分别为 TOC1、TOC2、TOC3、TOC4 和 TOC5。我们可以用输出比较功能对寄存器 TOC1 到 TOC5 中的值和自由运行计数器 TCNT 中的值进行比较。当 CPU 复位时，计数器从 $0000 开始计数，并且连续计数下去。当寄存器和计数器匹配时，相应的 OCx 标志位被置 1，从相关的输出引脚输出（如图 24-4 所示）。因此，通过对 TOCx 寄存器进行编程，就可以设定输出量出现的时间。这种输出比较功能能够产生比定时溢出标志更精确的时间延迟。

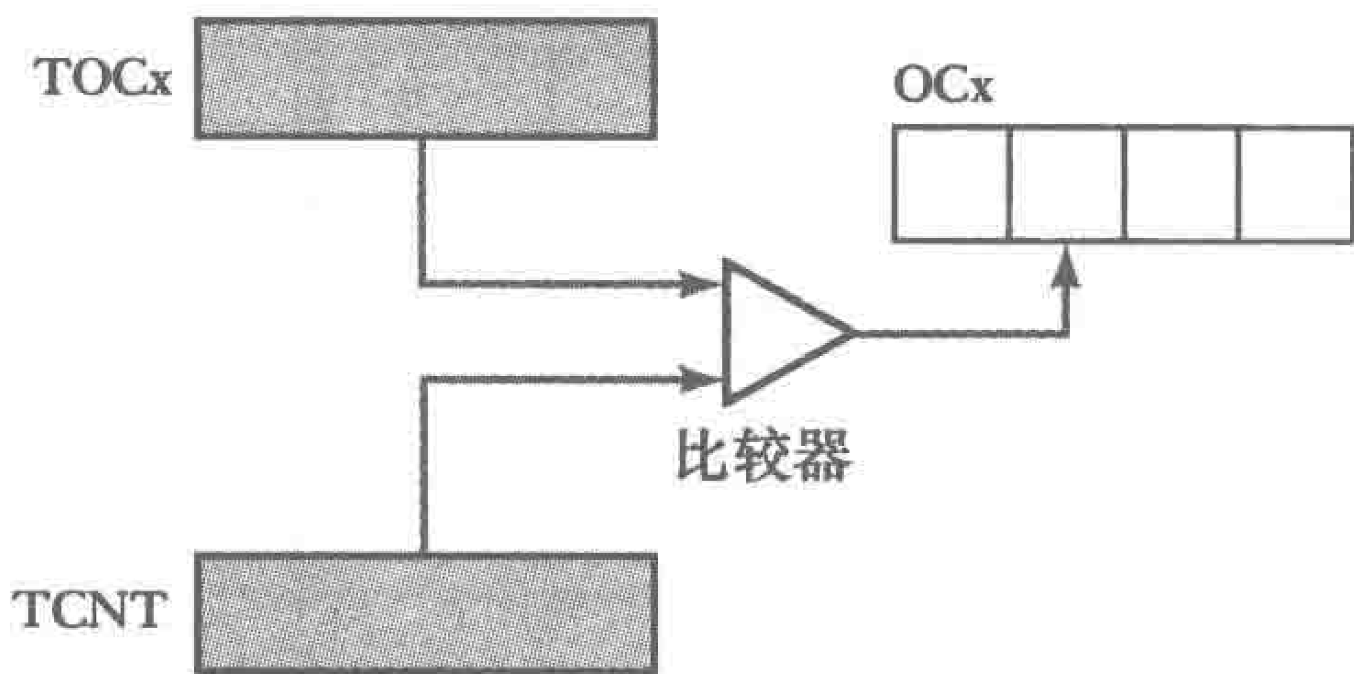


图 24-4　输出比较

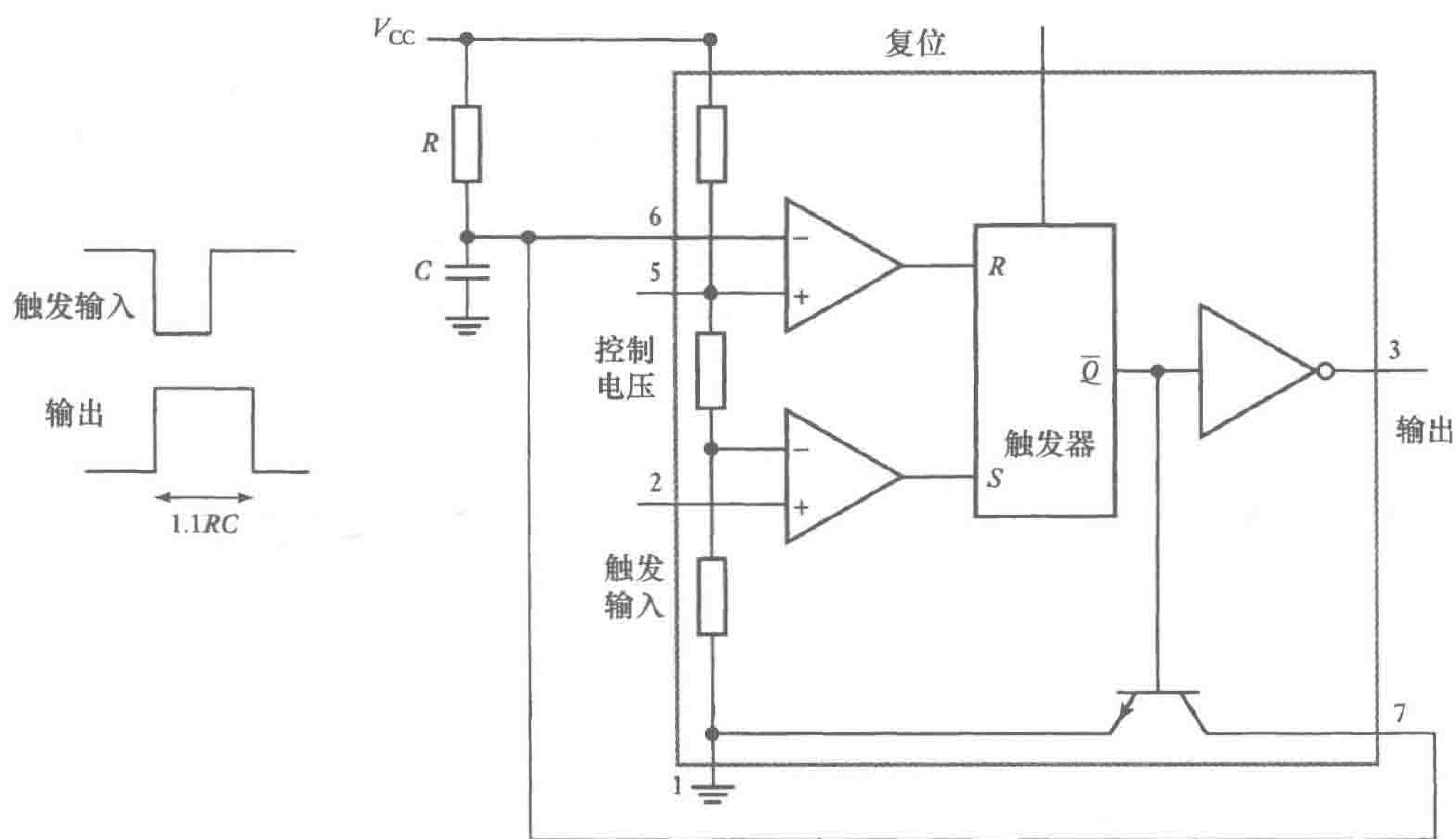
如下程序阐述了怎样用输出比较方法产生时间延迟。当 E 时钟的频率为 2MHz 时，输出比较方法能够产生的最长时间延迟是 32.7ms。为了能够产生更长时间的延时，需要多重输出比较操作。因此，我们可以使每个输出比较操作产生 25ms 的延时，然后重复 40 次，这样就可以达到 1s 的延时了。

```
REGBAS   EQU    $1000         ; 寄存器基地址
TOC2     EQU    $18           ; TOC2对REGBAS的偏移量
TCNT     EQU    $0E           ; TCNT对REGBAS的偏移量
TFLG1    EQU    $23           ; TFLG1对REGBAS的偏移量
OC1      EQU    $40           ; 屏蔽清零OCI脚和OCIF标志
CLEAR    EQU    $40           ; 清OC2F标志
D25MS    EQU    50000         ; E时钟周期数来产生一个25ms延时
NTIMES   EQU    40            ; 输出比较操作数给出1s延时
         ORG    $1000
COUNT    RMB    1             ; 为COUNT保留1个字节单元

         ORG    $C000
         LDX    #REGBAS
         LDAA   #OC1          ; 清零OC1标志
         STAA   TFLG1,X
         LDAA   #NTIMES       ; 初始化输出比较计数器
         STAA   COUNT
         LDD    TCNT,X
WAIT     ADDD   #D25MS        ; 加25ms延时
         STD    TOC2,X        ; 开始输出比较操作
         BRCLR  TFLG1,X OC1   ; 等待，直到OC1F标志置位
         LDAA   #OC1
         STAA   TFLG1,X       ; 清零OC1F标志
         DEC    COUNT         ; 输出比较计数器减1
         BEQ    OTHER         ; 如过了1s，则跳到OTHER
         LDD    TOC2,X        ; 准备开始一下一次比较操作
         BRA    WAIT
OTHER                         ; 程序的其他操作在1s延时后发生
```

另一种产生定时输出信号的方法就是使用定时器模块，如 555 定时器。对于 555 定时器，定时间隔是由外部电阻和电容设定的。图 24-5 给出了定时器以及当定时器被触发产生 on 输出状态所需的外部电路，on 输出状态的持续时间为 1.1RC。更大的持续时间需要更大的 R 和 C 值。电阻 R 值限定在 1MΩ 左右，否则会有泄漏问题；电容 C 的值被限制在 10μF 左右，这样电解电容泄漏和低精度问题就能避免了。因此图 24-5 中电路的延迟时间小于 10s。电阻和电容的较低限制是 $R=1\text{k}\Omega$，$C=100\text{pF}$，延迟时间小于 1ms。对于更长时间的定时，如从 16ms 到数天，可以选择 ZN1034E 等可替换的定时器。

538 ~ 539



540

图 24-5 555 定时器

24.1.2 刮水器运动

考虑一种设备，要求其摇臂能够像刮水器一样以一定的弧度前后摆动。可能的解决方案是：

1）机械连杆和直流电动机。

2）步进电动机。

机械解决方案如图 24-6 所示。电动机驱动摇臂 1 旋转，使摇臂 2 传递给摇臂 3 一个摇摆运动。汽车刮水器普遍采用这种直流永磁电动机驱动的机构。

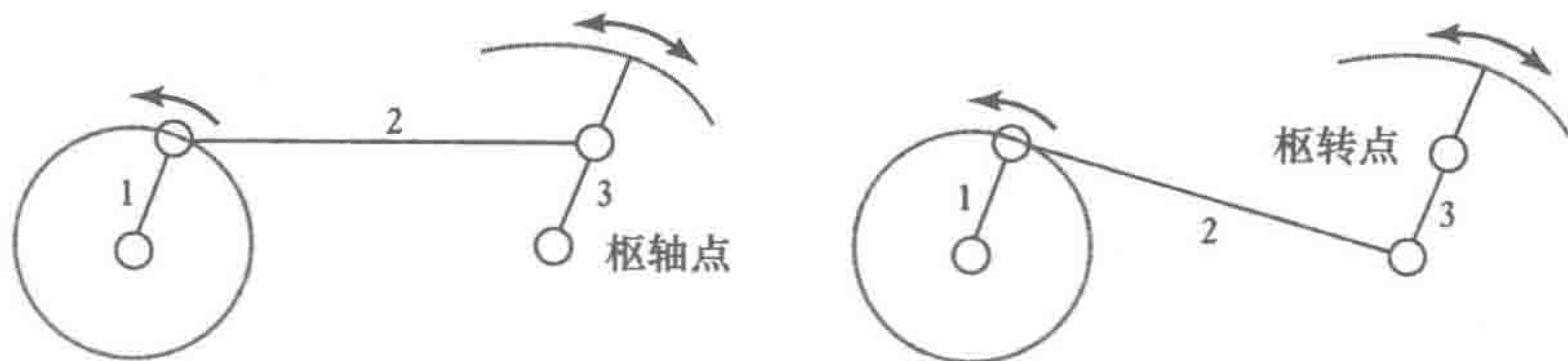


图 24-6 刮水器机构

另一种可选的解决方案是采用步进电动机。图 24-7 给出了一个有外围接口适配器（PIA）的微处理器或一个微控制器如何用于步进电动机。步进电动机的输入必须能使电动机朝一定方向旋转一定步数，然后再反方向旋转相同的步数。

如果步进电动机处于“整步”方式，则输出如表 24-1a 所示。因此启动并使电动机向前旋转就要涉及 A、9、5、6 的顺序，然后再从 A 开始。若要反向旋转可以采用 6、5、9、A 的顺序，然后再从 6 开始。如果步进电动机采用“半步”方式工作，输出如表 24-1b 所示。当顺序是 A、8、9、1、5、4、6、2 时，向前运动，然后又回到 A；向后运动时为 2、6、4、5、1、9、8、A，然后又回到 2。

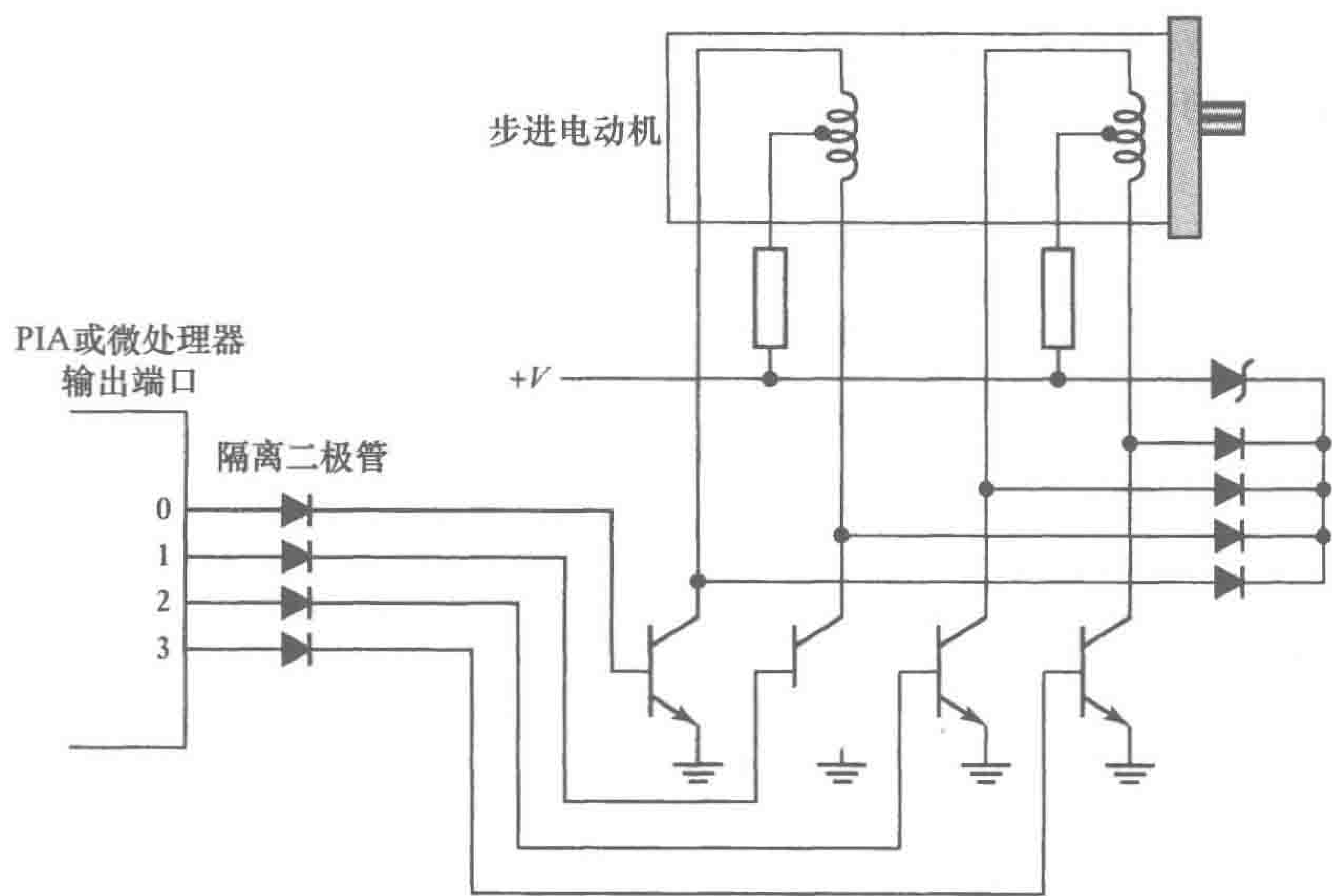


图 24-7 步进电动机的接口

表 24-1

a）整步

步	位 3	位 2	位 1	位 0	代码
1	1	0	1	0	A
2	1	0	0	1	9
3	0	1	0	1	5
4	0	1	1	0	6
1	1	0	1	0	A

b）半步配置

步	位 3	位 2	位 1	位 0
1	1	0	1	0
2	1	0	0	0
3	1	0	0	1
4	0	0	0	1
5	0	1	0	1
6	0	1	0	0
7	0	1	1	0
8	0	0	1	0
1	1	0	1	0

程序的基本组成部分为：

1）前进一步
2）跳到延时程序，给出完成步进所需要的时间
3）循环或重复上一步直到达到前进方向要求的步数
4）改变方向
5）重复上面的步骤，向反方向前进相同的步数

为了实现3“半步”向前，3 步向后，C 语言程序可包括如下部分（在包含合适的头文件之后）：

```
main ( )
{
   portB = 0xa; /* 第1步*/
   delay ( ); /*结合延时程序，如20ms*/
   portB = 0x8; /* 第2步*/
   delay ( ); /* 结合20ms延时程序*/
   port B = 0x9; /* 第3步*/
   delay ( ); /* 结合20ms延时程序*/
   port B = 0x8; /*返向一步*/
   delay ( ); /* 给20ms延时程序*/
   port B = 0xa; /*返向一步*/
   delay ( ); /* 结合20ms延时程序*/
   port B = 0x2; /* 返回到电动机开始的位置*/
   delay ( ); /* 结合20ms延时程序*/
}
```

当涉及很多步数时，一个更简单的程序需要使用计数器。每进行一步，计数器增加1，循环，直到计数器中的值达到所需步数。这样程序的基本形式如下：

541~542

1)前进一步。
2)跳到延时程序处,给出完成步进所需要的时间。
3)计数器加1。
4)连续循环或重复以上步骤,直到计数器达到前进方向所需步数为止。
5)改变方向。
6)重复上面的步骤,向反方向前进相同的步数。

集成电路用于步进电动机控制可以简化接口和软件。这种电路的使用如图24-8所示。电动机的步进是由一个高-低-高脉冲的低到高的跳变产生的，所需要的是向触发器（trigger）输入脉冲的数量。向旋转端（rotation）输入高电平，使电动机逆时针旋转；输入低电平，使电动机顺时针旋转。因此，只需要来自微处理器的输出信号作为到触发器的输出脉冲和一个输出到“旋转端”的信号。输出到置位端的信号用于将电动机恢复到初始位置。

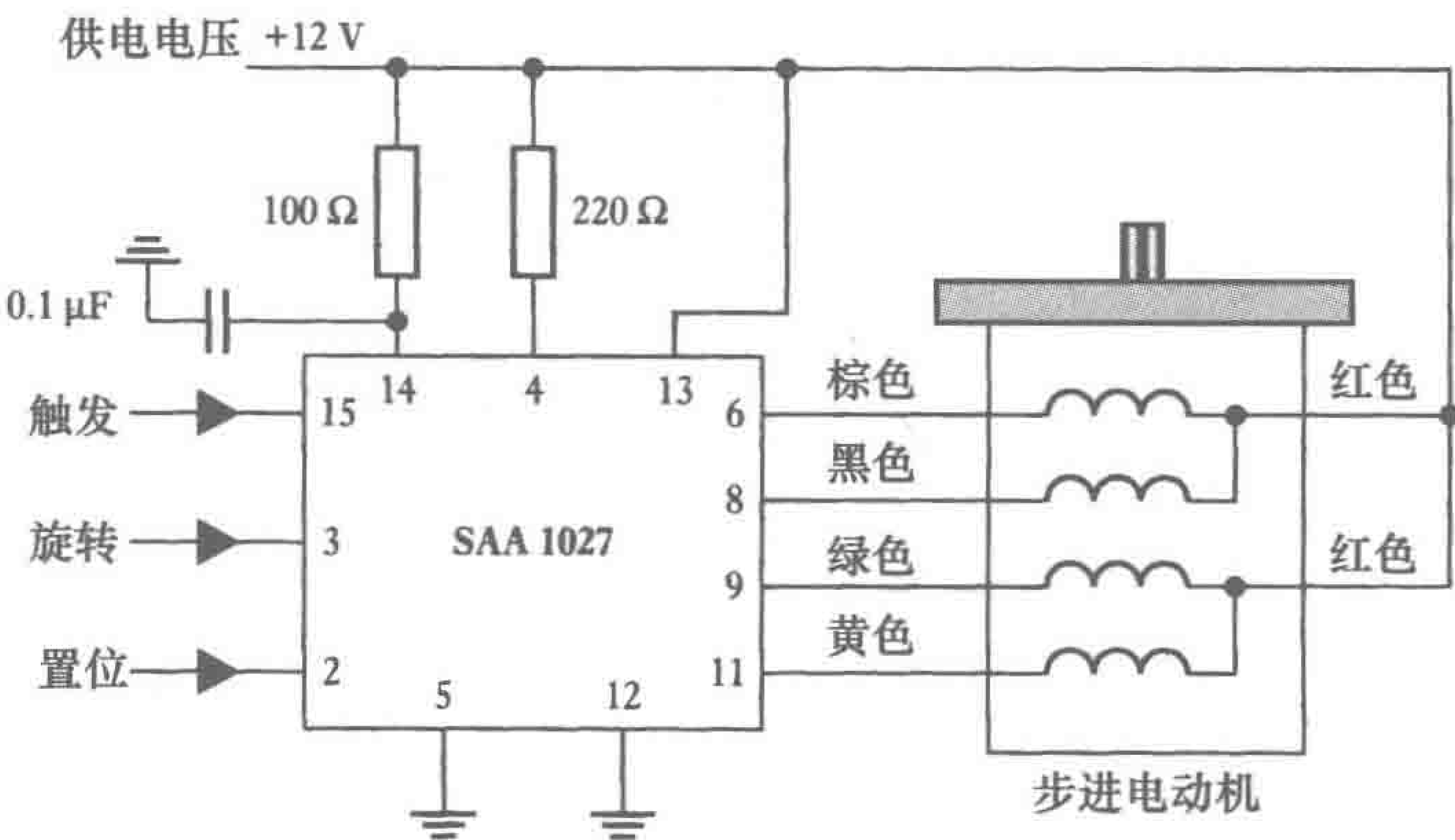


图24-8　适用于步进电动机的集成电路SAA1027

以上解释了如何用步进电动机产生角位移，但是当电动机输入为电压信号时，电动机将如何反应呢？我们能否期望电动机在达到所需角度前没有角度超调和振荡？为解释如何为步进电动机系统研发一种运动模型并预测它的行为，考虑如下简化分析（欲了解更多细节，请参考《Stepping Motor and their Microprocessor Controls》，作者为T. Kenjo和A. Sugawara（Clarenden Press，1995））。

这种系统是一种开环控制系统，系统包括一个由微控制器的脉冲驱动的步进电动机。永磁步进电动机（见9.7节）有一个带有多个磁极的定子，磁极由缠绕在上面线圈中的电流激励。当有电压脉冲输入时，我们为转子旋转确定一种模型，为了简单，我们只考虑有一对磁极的步进电动机，并且和18.3.2节中分析的直流电动机处理方式一样。如果v是提供给电动机线圈对的电压，v_b为反电动势，则

$$v - v_b = L\frac{di}{dt} + Ri$$

其中L为电路的电感，R为电阻，i为电路电流。假设电感值不会发生明显的改变，因此假定L为常量。

543

反电动势与线圈中通过的磁通量的变化率成比例。还依赖于转子与相关磁极的角度θ。因此，可以得到：

$$v_b = -k_b\frac{d}{dt}\cos\theta = k_b\sin\theta\frac{d\theta}{dt}$$

其中 k_b 是常量，因此

$$v - k_b \sin\theta \frac{d\theta}{dt} = L\frac{di}{dt} + Ri$$

对上式进行拉普拉斯变换，可得：

$$V(s) - k_b s \sin\theta\,\theta(s) = sLI(s) + RI(s) = (sL + R)I(s)$$

与直流电动机一样，通过线圈中的电流将会产生一个转矩（由牛顿第三定律可知：作用到磁性转子的转矩即为施加在定子线圈上的反作用转矩）。这个转矩与线圈中的磁通密度和电流的乘积成比例。而磁通密度则依赖于转子的角位置，所以我们可以写为：

$$T = k_t i \sin\theta$$

其中 k_t 为常量。该转矩会引起一个角加速度 α，又由于 $T = J\alpha$，J 为转子的转动惯量，所以

$$T = J\frac{d^2\theta}{dt^2} = k_t i \sin\theta$$

将上式进行拉普拉斯变换，可得：

$$s^2 J\theta(s) = k_t \sin\theta I(s)$$

可以得出：

$$V(s) - k_b s \sin\theta\,\theta(s) = (sL + R)(s^2 J\theta(s)/k_t \sin\theta)$$

输入电压和输出角位移的传递函数为：

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{k_t \sin\theta}{J(sL + R)s^2 + k_b k_t s \sin^2\theta}$$

$$= \frac{1}{s} \times \frac{k_t \sin\theta}{JLs^2 + JRs + k_b k_t \sin^2\theta}$$

当给电动机电压脉冲时，因为对于单位电压脉冲 $V(s) = 1$，所以，

$$\theta(s) = \frac{1}{s} \times \frac{k_t \sin\theta}{JLs^2 + JRs + k_b k_t \sin^2\theta}$$

$$= \frac{1}{s} \times \frac{(k_t \sin\theta)/JL}{s^2 + (R/L)s + (k_b k_t \sin^2\theta)/JL}$$

S 的二次方程是 $s^2 + 2\zeta\omega_n s + \omega_n^2$（见 20.3.1 节）的形式。因此，我们可以得到一个自然频率 $\omega_n = \sqrt{\frac{k_b k_t \sin^2\theta}{JL}}$ 和阻尼系数 $\zeta = \frac{R}{2L\omega_n}$。转子将旋转一定的角度并在这个角度附近振荡，振荡随时间逐渐消失。

汽车挡风玻璃雨刮器一般是由一台直流电动机驱动的连杆机构。在维基百科（Wikipedia）网站上有关于雨刮器系统的操作演示讨论。

24.1.3　浴室秤

考虑一个简单浴室秤的设计。主要的要求是：当人站在平台上时，人的体重会以某种可读形式显示出来。人的体重应该以合理的速度和精确度给出，而且人的体重与人站在平台上的位置无关。可能的解决方案包括：

1）基于齿轮和弹簧的纯机械系统。

2）一个称重传感器和一个微处理器或者微控制器系统。

一种可行的解决方法是用平台上人的体重使两片平行放置的弹簧叶片发生弯曲（如图 24-9a 所示）。在这种情况下，弹簧的弯曲与人站在平台上的位置就几乎无关了。通过图 24-9b 所示的装

置就可以将弹簧的弯曲运动转换为秤上指针的运动。齿轮齿条机构可以将直线运动转换为绕水平轴的旋转运动，然后通过锥齿轮又可将水平轴的旋转运动转换为竖直轴的旋转运动，最后带动指针运动。

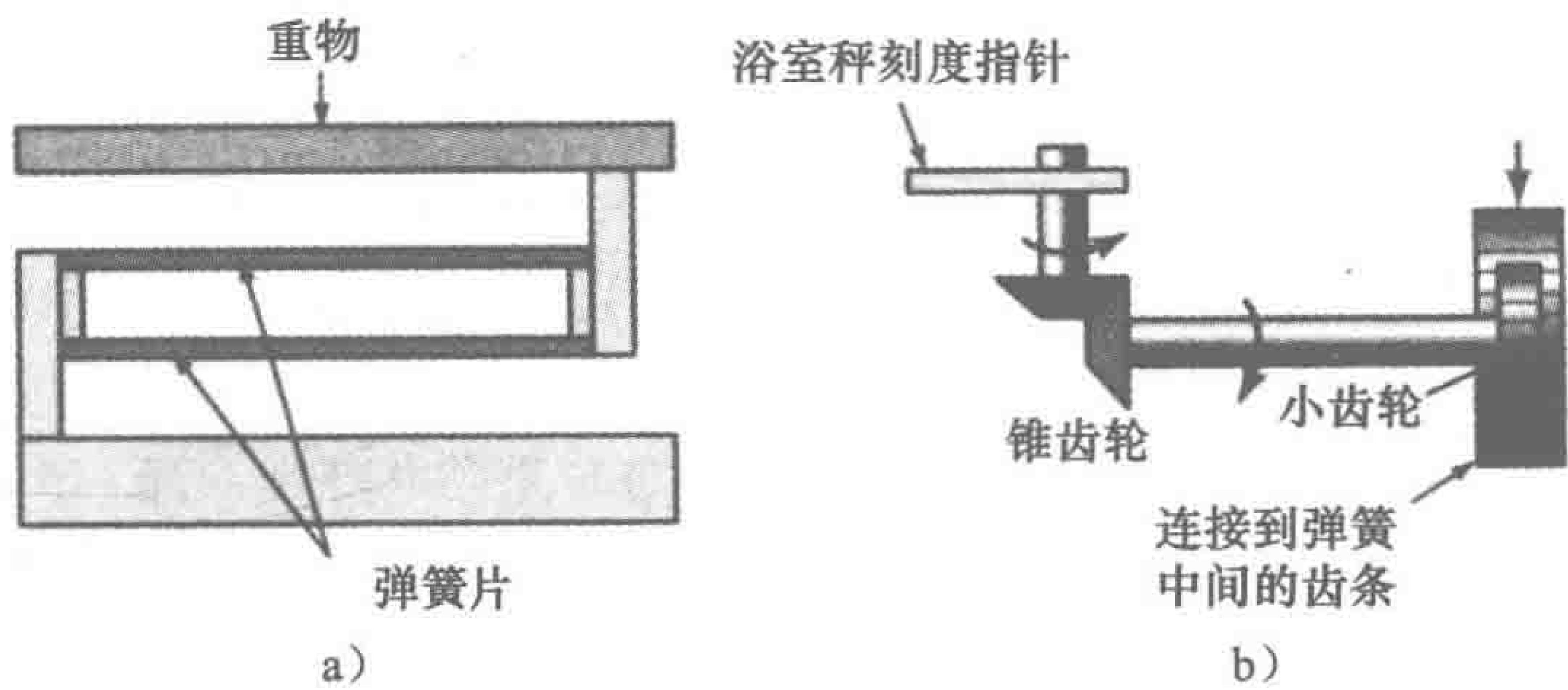


图 24-9 浴室秤

另一种可行的解决方法使用微处理器。称重平台可以固定在采用了电阻应变计的称重传感器上。当人站在平台上时，应变片受到应力，电阻值发生改变。如果应变片固定在惠斯通电桥的四个臂上，那么电桥的不平衡电压输出就是人体重的一种度量。这个不平衡电压可以通过一个微分运算放大器进行放大。由此产生的模拟信号可以通过一个有锁存功能的 AD 转换器输入到微处理器（例如 Motorola 6820）中。图 24-10 给出了输入接口。此外，还需要一个非易失性的存储器，这可用可擦写可编程只读存储器（EPROM）芯片，如 Motorola 2716。显示输出可通过外围接口适配器得到（PIA，如 Motorola 6821）。

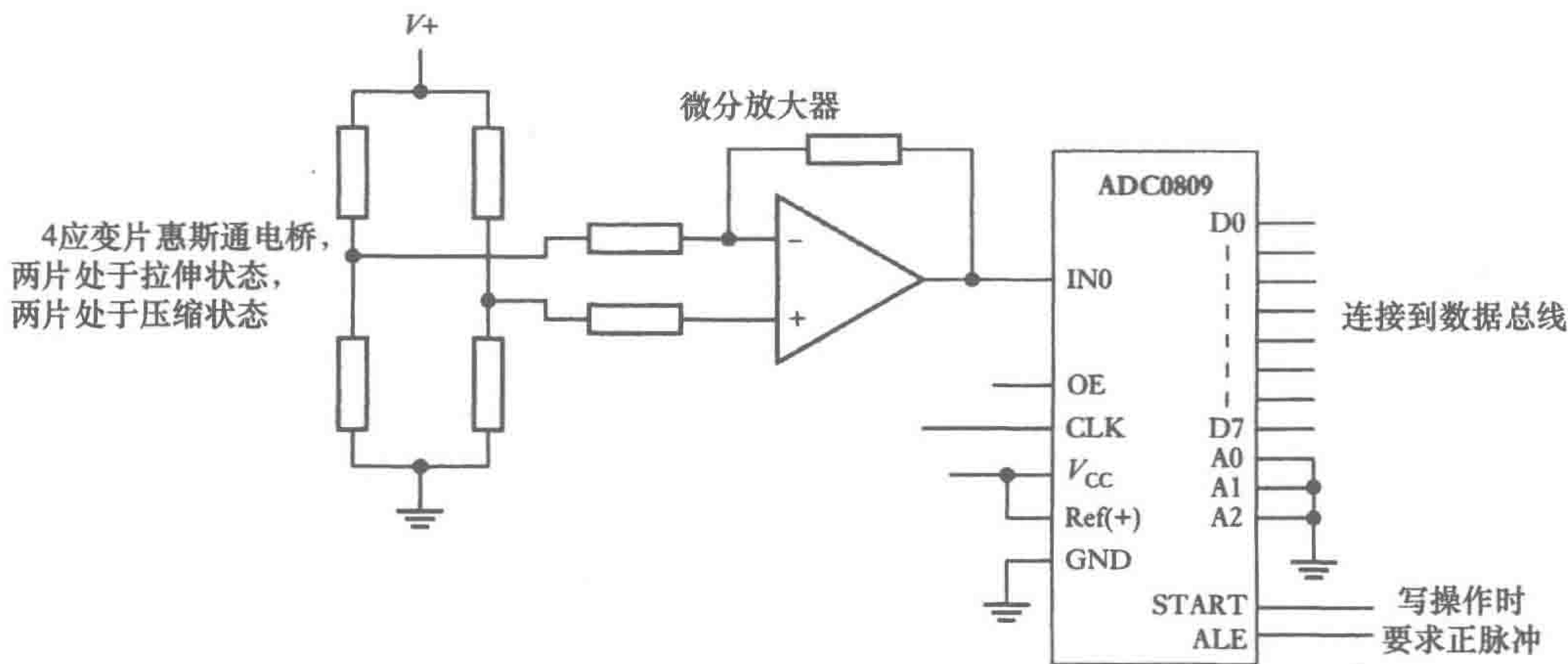


图 24-10 输入接口

然而，如果采用微控制器，则存储器包含在单个控制器芯片中。采用合适的微控制器，如 M68HC11，我们可以获得输入信号的模数转换。系统从而变成：应变片输出的电压信号通过微分运算放大器之后输入到微控制器的端口 E（ADC 输入），输出信号经过合适的驱动后从端口 B 和端口 C 输出到译码器，然后再输出到 LED 显示器（如图 24-11 所示）。

程序结构为：

```
清空 LED 显示器和存储器,初始化
开始
    有人站在称上? 如果没有,显示 000
    如果有
      输入数据
```

将重量数据转换为合适的输出形式
输出到译码器,LED 显示
延时,保持显示数据
从开始处重复检测

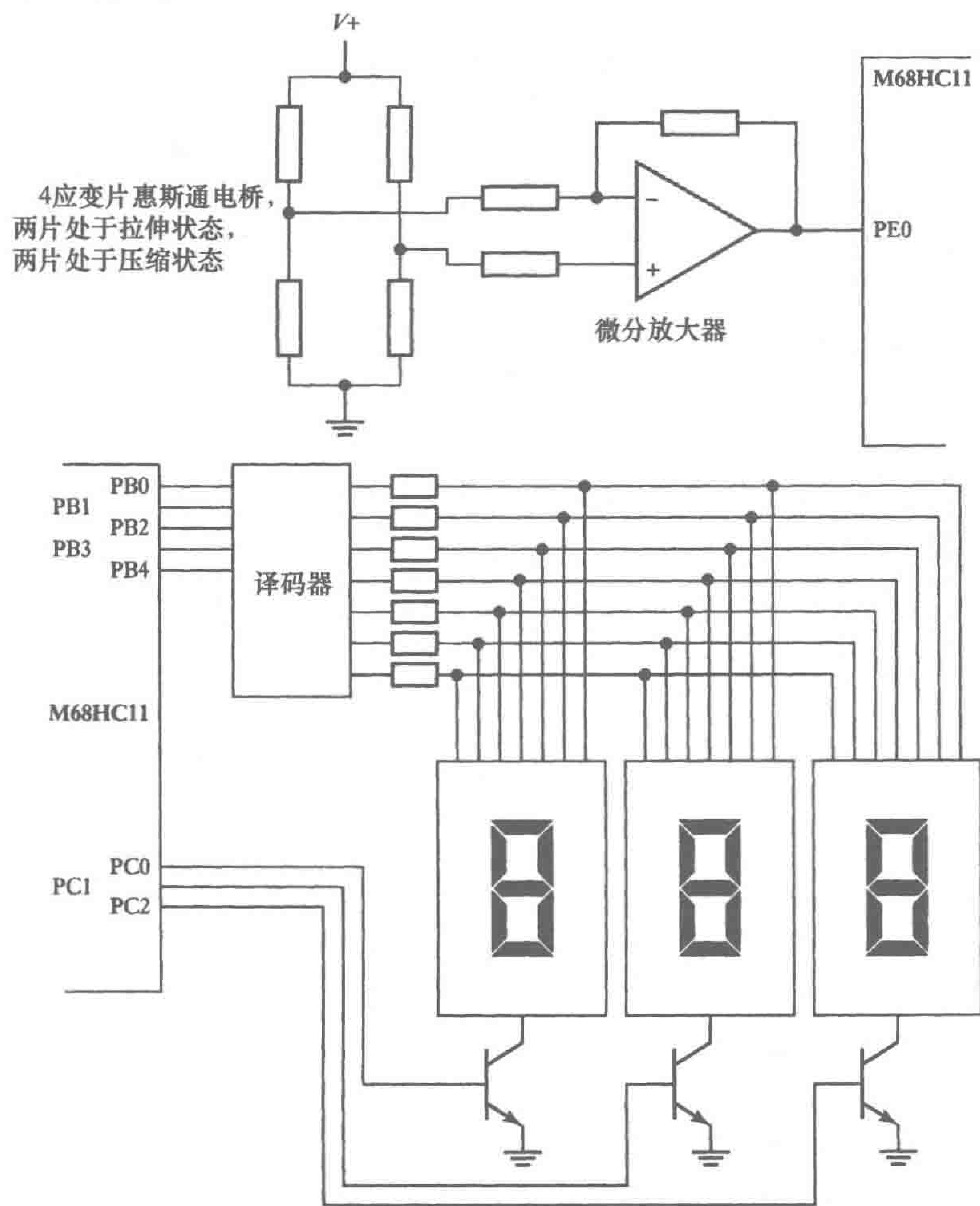


图 24-11　浴室秤

对于浴室秤的机械部分设计来说，我们需要考虑当人站在秤上时会发生什么？与图 17-3a中描述的系统相比（见 17.2.2 节），我们有一个弹簧阻尼质量系统，可用下面的方程来描述它的运动：

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2}+c\frac{\mathrm{d}x}{\mathrm{d}t}+kx=F$$

其中，x 是当有力 F 作用时平台的垂直变形量，将上式进行拉普拉斯变换，可得：

$$ms^2X(s)+csX(s)+kX(s)=F(s)$$

于是描述系统的传递函数可以写成如下形式：

$$G(s)=\frac{X(s)}{F(s)}=\frac{1}{ms^2+cs+k}$$

我们可考虑一个重 W 的人站在平台上相当于有一个阶跃输入，则

$$X(s)=\frac{1}{ms^2+cs+k}\times\frac{W}{s}$$

s 的二次方程也是 $s^2+2\zeta\omega_n s+\omega_n^2$ 这种形式（见 20.3.1 节），所以有自然频率 $\omega_n=\sqrt{\frac{k}{m}}$，阻尼因子 $\zeta=\frac{c}{2\sqrt{mk}}$。

当人站在秤上时，总希望自己的体重能够快速显示，而不是围绕体重值振荡很长时间。如果把阻尼调到临界值，则需要很长的时间才能显示体重值，所以，一般将阻尼值调节的有些振荡，但这些振荡会很快消失。一般达到稳定值上下 2% 内所用的建立时间 t_s 是适当的（见 19.5 节），比如 4s。因为 $t_s=\frac{4}{\zeta\omega_n}$，所以要求 $\zeta\omega_n=1$，$\zeta=\sqrt{\frac{m}{k}}$。因此，改变 ζ 的一种简单方法就是改变质量。

544~547

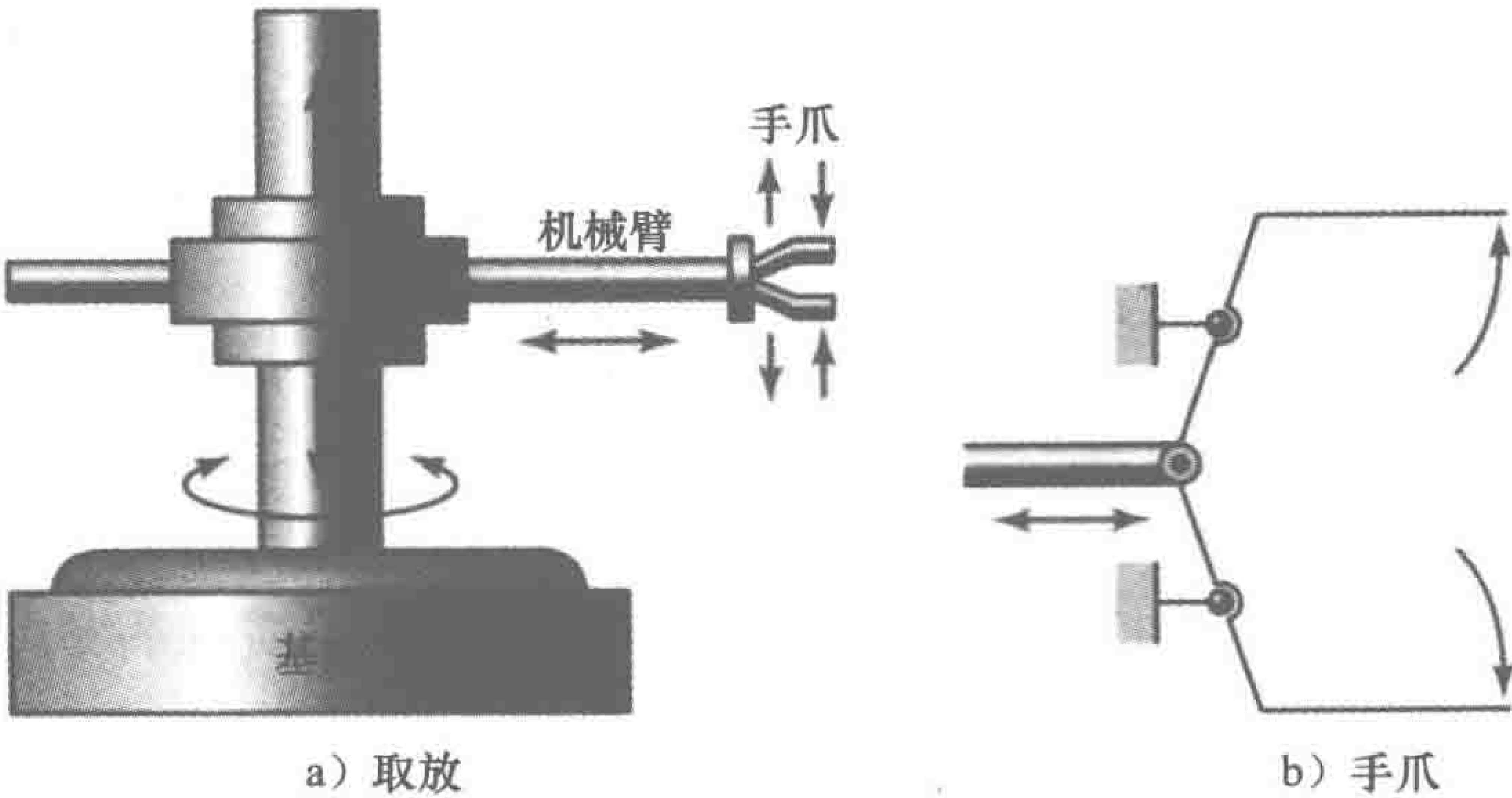
以上说明了怎样用数学模型来预测系统的行为，并通过改变哪些因素可以改善系统的性能。

24.2　案例研究

以下是一些机电一体化系统例子的概要介绍。

24.2.1　取放机器人

图 24-12a 给出了一个取放机器人单元的基本形式。机器人有三根轴，可以实现如下运动：基座的顺时针或逆时针旋转，机械臂的伸长或缩短以及上升或下降，还有手爪的开或合。这些运动由气缸驱动，而气缸则是由带有限位开关的电磁阀控制，限位开关可以表示一个动作已经完成。因此，装置的顺时针旋转运动是由气缸中活塞的伸展引起的，逆时针旋转是由活塞的缩回产生的。同样，机械臂的上升运动是由直线气缸中活塞的伸展产生的，机械臂下降是由活塞的缩回引起的；机械臂的伸长是由另一个气缸中活塞的伸展产生的，机械臂的缩回是由该气缸中活塞的缩回产生的。手臂的开或合是由直线气缸中活塞的伸展或缩回产生的。图 24-12b 展示了可能用到的手爪基本机构。



a）取放　　　　b）手爪

图 24-12　一款机械臂

这种机器人的典型程序如下：

1)在上面给料机中悬挂的组件上,竖直闭合手爪。
2)缩回手臂,使组件从给料机中取出。
3)在水平面内旋转机械臂使它的方向朝向工件。
4)伸展机械臂,使手爪在工件上方。
5)旋转手腕,使组件从手爪向下垂。
6)释放手爪,使得组件落入要求的位置。
7)旋转手爪,使其达到竖直向上位置。

8)收缩机械臂。

9)旋转机械臂指向给料机。

重复以上顺序取下一个组件。

图 24-13 说明了如何用微控制器控制电磁阀，进而控制机器人单元的运动。 548

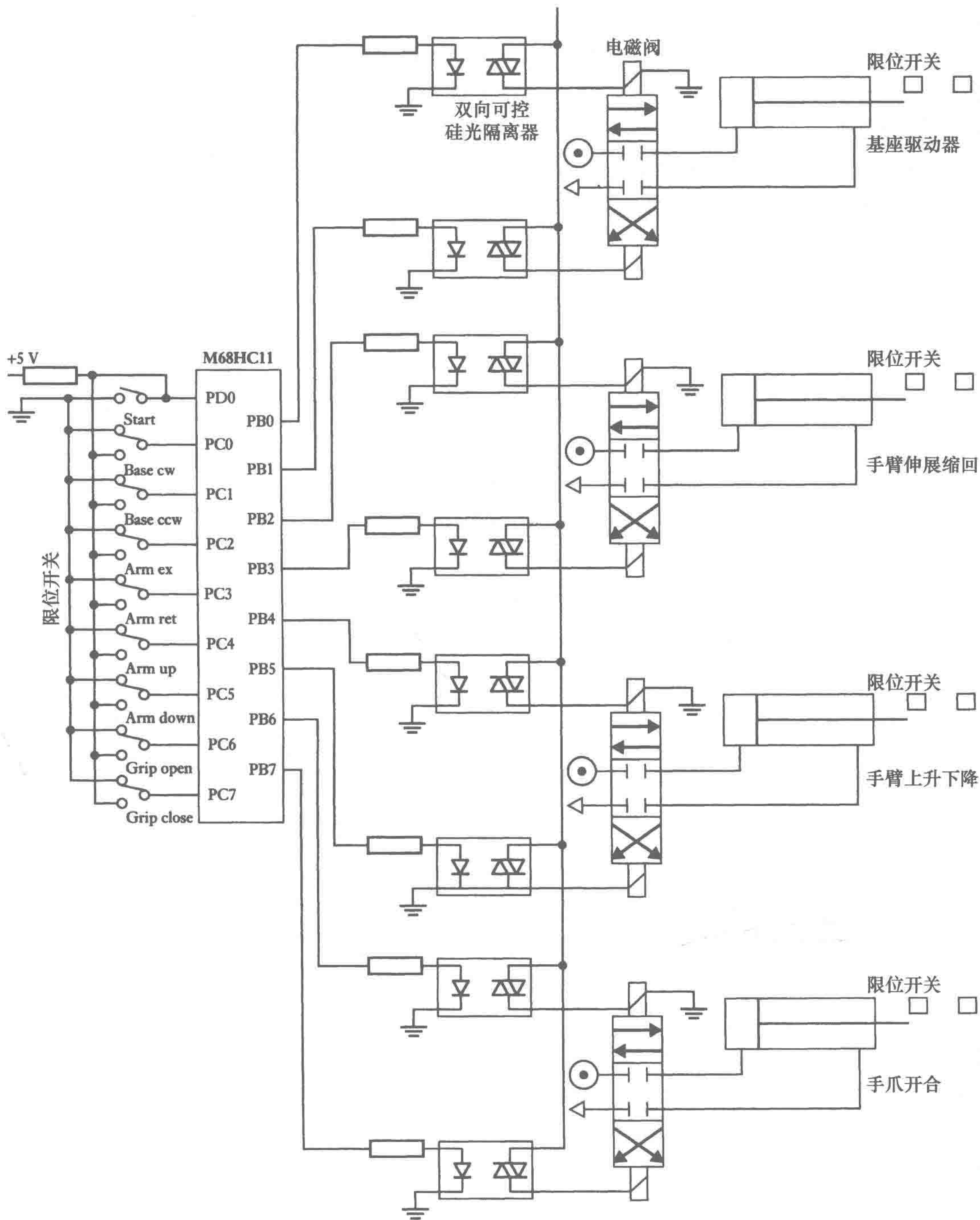


图 24-13 机器人控制

液压和气压活塞广泛应用于机械臂的驱动，因为它们可以很容易地控制手臂以相对较低的速度移动，而电动机驱动则需要用到减速器。

在图 24-13 中，机械臂和手爪的位置由限位开关决定。这就意味着每个驱动器只有两个位置可以精确达到，而且除非限位开关的物理位移可以移动，否则位置是不能轻易改变的。上述电路构成了一个开环控制系统。在某些应用中，开环是不会出现问题的。

然而，通常都是用闭环控制系统来控制机械臂和手爪，而且机械臂和手爪的位置由传感器检测并将信息反馈到控制器中与所要求的位置相比较。当与所要求的位置有偏差时，控制器通过控制驱动器的运动来减少误差。关节的角位置通常由编码器检测（见 2.3.7 节），这能达到很高的精度。图 24-14 所示的闭环控制系统可用于控制机器人手臂的线性运动。

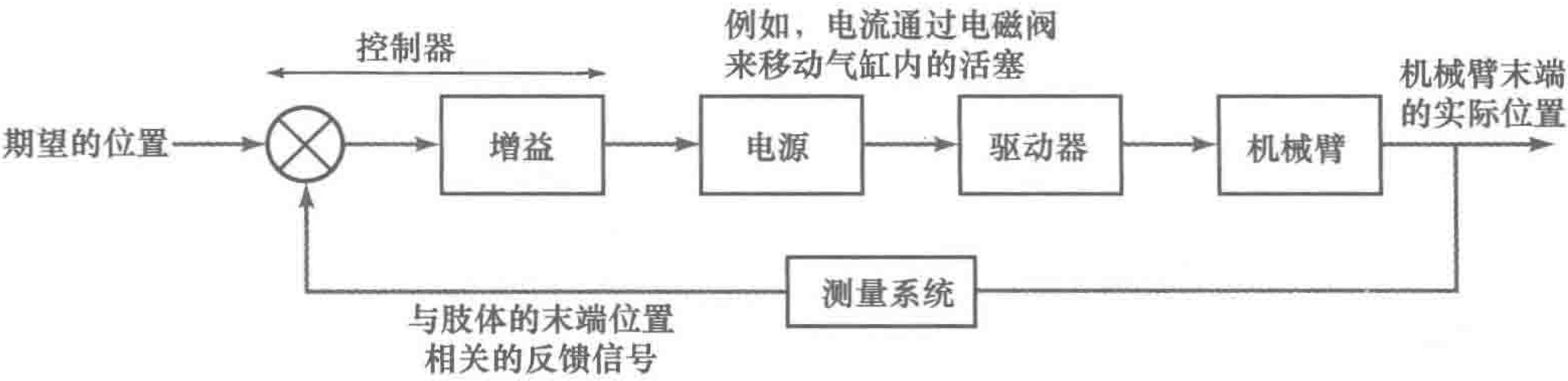


图 24-14 机械臂闭环控制

驱动器的输出是一个作用于手臂末端的力 F。对于一个设定的位置 y_s 以及实际达到的位置 y，误差信号为 y_s-y，为了简化假定测量系统的增益为 1。如果控制器增益为 G_c，驱动器组件的增益为 G_a，则力 $F=G_cG_a(y_s-y)$。被加速的质量为机械臂携带的负载质量、机械臂的质量以及驱动器中移动零部件的质量。如果总质量为 m，则由牛顿定律可得 $F=ma$，其中加速度 a 可以写成 $\mathrm{d}^2y/\mathrm{d}t^2$。然而，这里并没有考虑摩擦力，又因为我们可以认为摩擦力与速度成比例，摩擦力为 $k\mathrm{d}y/\mathrm{d}t$。因此，可以得到：

$$F = G_cG_a(y_s - y) = m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + k\frac{\mathrm{d}y}{\mathrm{d}t}$$

所以，

$$y_s = \frac{m}{G_cG_a}\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + \frac{k}{G_cG_a}\frac{\mathrm{d}y}{\mathrm{d}t} + y$$

这是一个二阶微分方程，其中 y 和 20.3.1 节所描述的一样，而它的形式则依赖于阻尼因子。而欠阻尼系统有一个自然角频率。

549~550

$$\omega_n = \sqrt{\frac{G_cG_a}{m}}$$

这个角频率决定了系统对变化的响应速度（见 19.5 节）：角频率越大，系统响应越快（上升时间与这个角频率成反比）。这意味着增大控制器的增益或者减少系统的质量都能增大系统的响应速度。阻尼系数 ζ 由微分方程可得。

$$\zeta = \frac{k}{2\sqrt{G_cG_am}}$$

振荡消失的时间，也就是建立时间（见 19.5 节），是与阻尼因子成反比的，所以增加任何一部分的质量都会导致阻尼因子的减小，就需要更长的时间振荡才能消失。

24.2.2 停车栏杆系统

为了更好地解释 PLC 的用法，现在我们来看停车场的投币停车系统。当足够的硬币投入投币箱时，驶入栏杆打开；当在停车场栏杆一侧检测到车辆时，驶出栏杆打开。图 24-15 给出了用于升、降栏杆的阀门系统的种类。

当电流流经电磁阀 A 时，缸体中的活塞向上移动，使栏杆绕转轴旋转提升，从而让车辆通过；当流经电磁阀 A 的电流停止时，阀门的返回弹簧使阀门回到初始位置。当电流流入电磁阀 B 时，压力使得栏杆下降。限位开关用于检测栏杆的下降以及栏杆的完全上升。

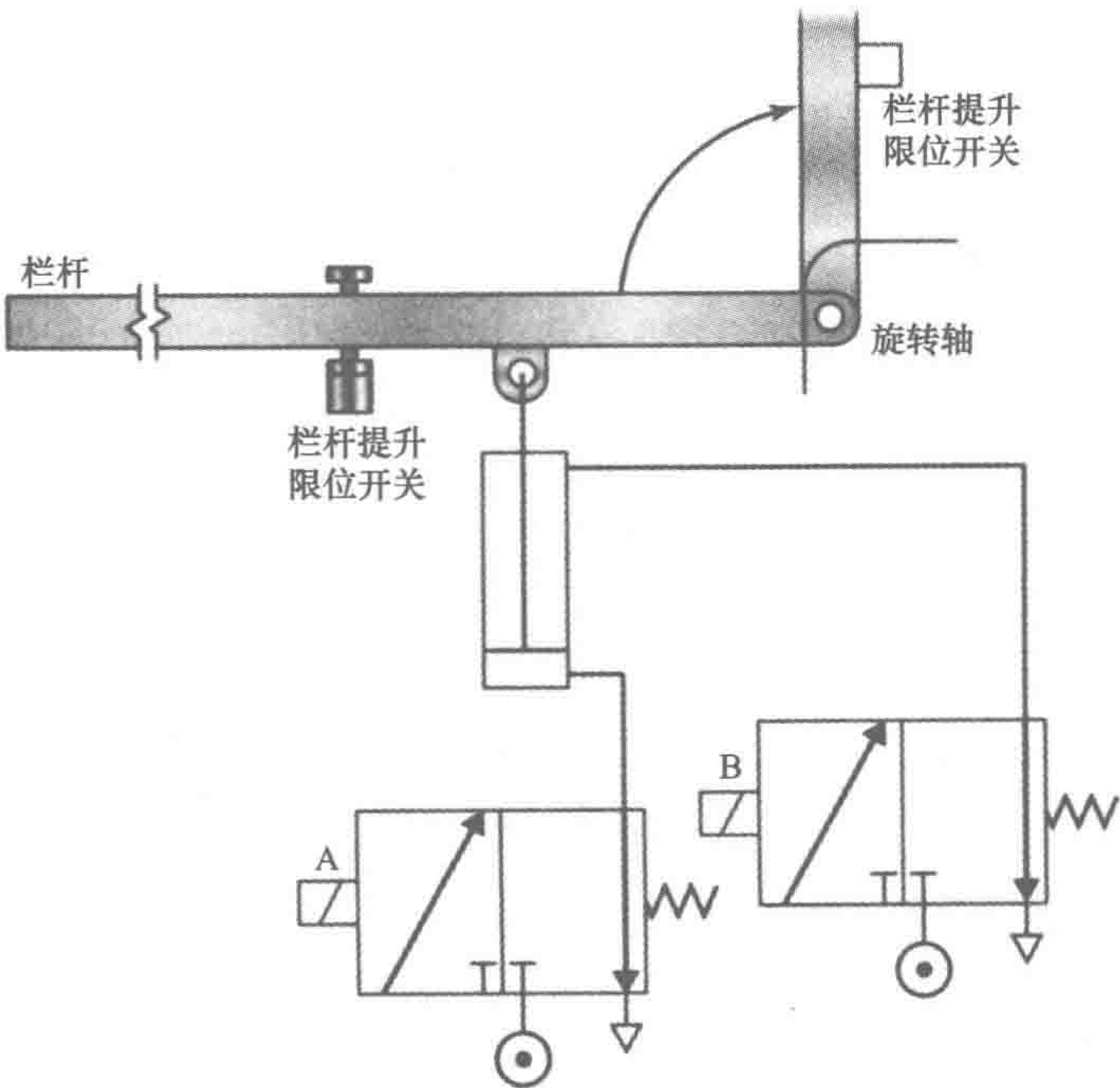


图 24-15 栏杆提升下降系统

551

图 24-15 所示的两个系统，一个用于控制车辆的驶入，一个用于控制车辆的驶出。图 24-16 给出了与 PLC 输入/输出点的连接图，图 24-17 为梯形图程序。

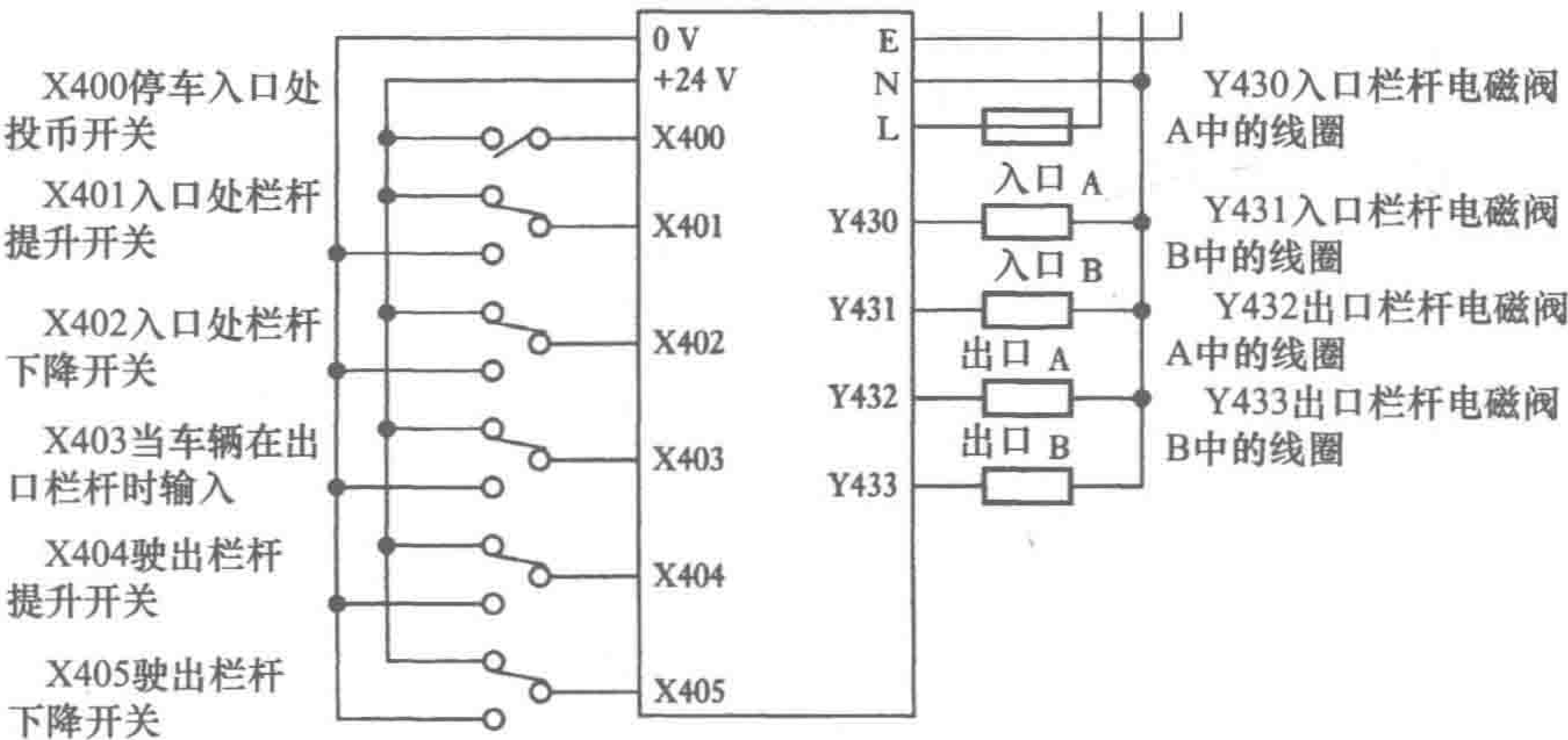


图 24-16 PLC 连接

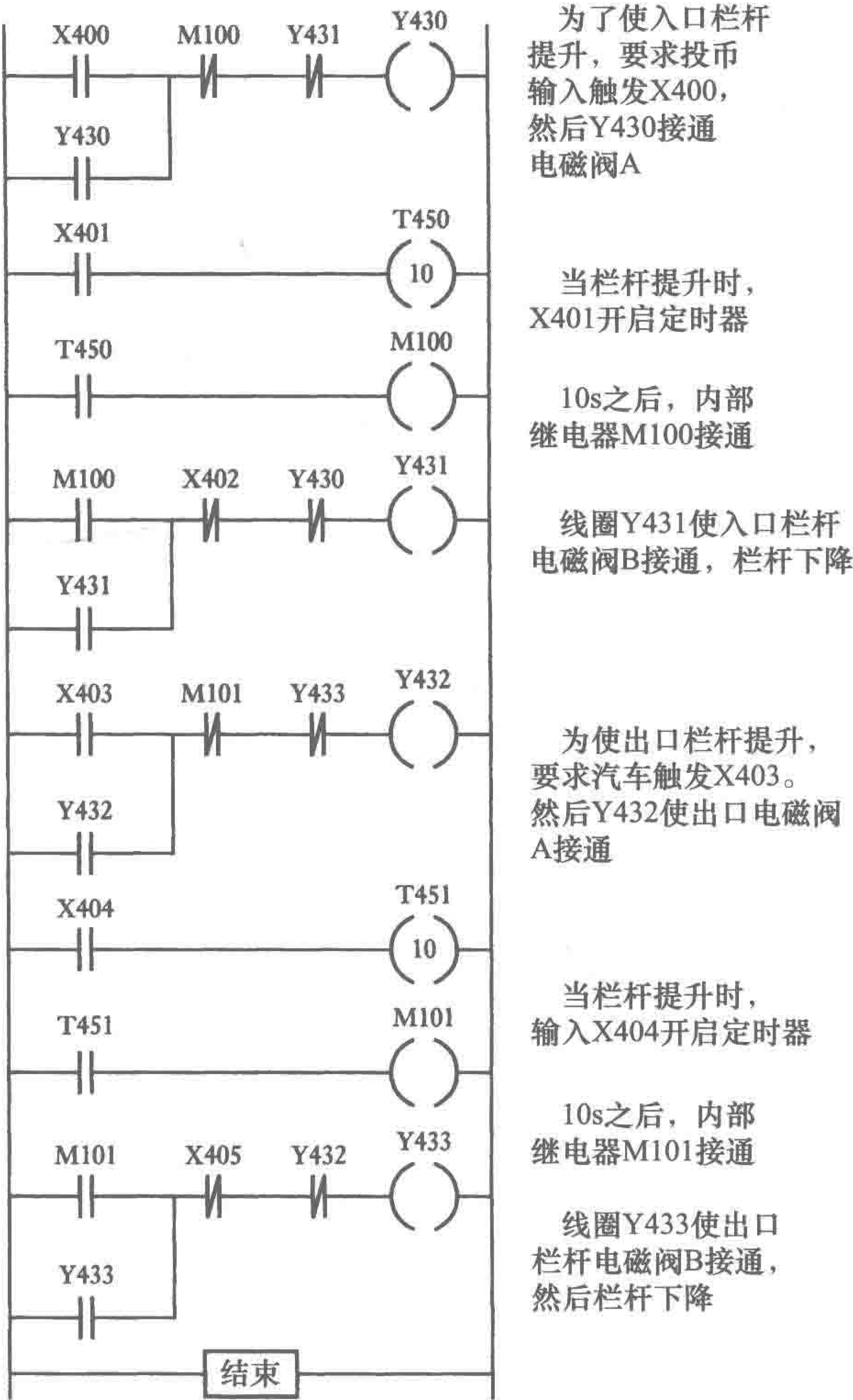


图 24-17　梯形图

24.2.3　数码相机

数码相机就是捕捉图像并将其以数字形式存储在存储卡中。它和先前的胶片相机不同，胶片相机中的图像是以化学变化这种模拟形式存储在胶片上。图 24-18 给出了一个比较便宜的数码相机的基本元件。

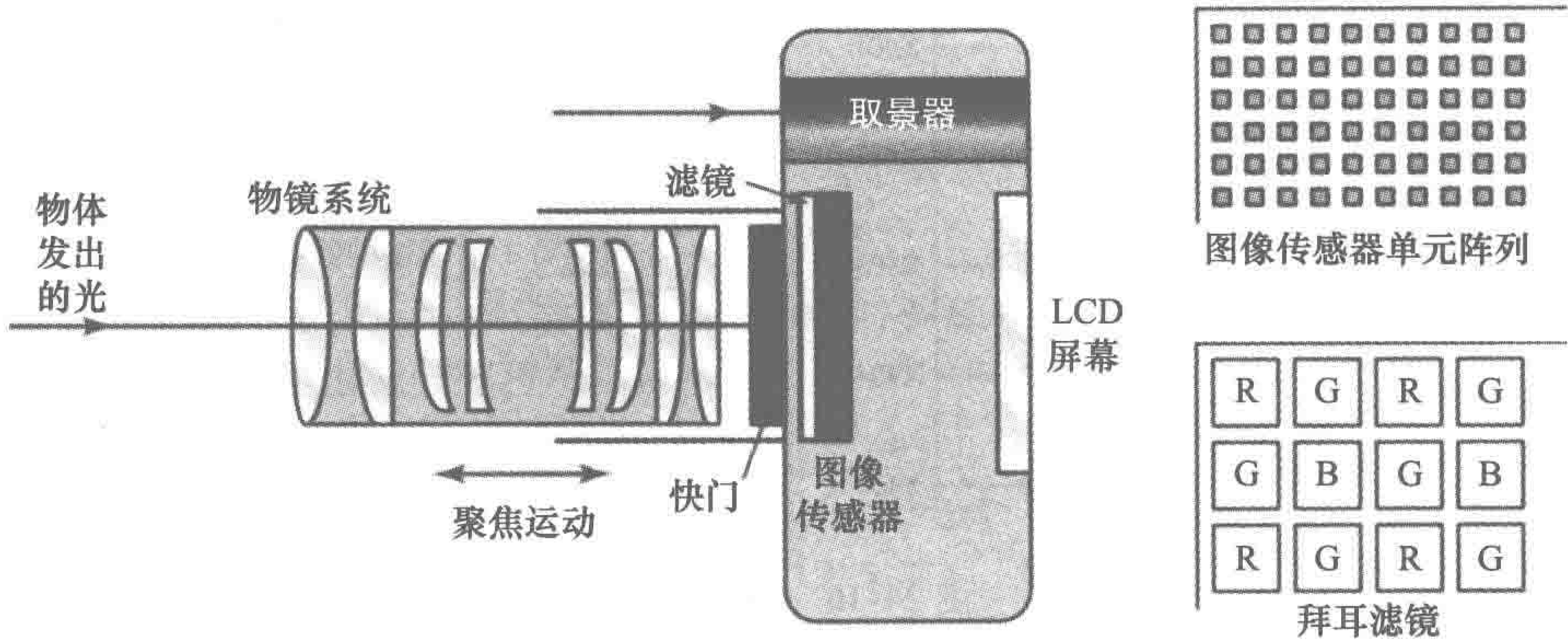


图 24-18　数码相机的基本组成

当摄影师不完全地按下快门按键到第一个位置时，微控制器以测量传感器的输入计算快门的速度和光圈设定，并将它显示在液晶显示屏（LCD）上。同时，微控制器对距离传感器的输入进行处理，并发出信号驱动电动机调节镜头的焦距。当摄影师完全按下快门到第二个位置时，微控制器发出信号将光圈调整到所需大小，并且打开快门获得所需的曝光时间。然后，当快门关闭时，对图像传感器接收到的图像进行处理，并将它存储在存储卡上。而且，当快门按键不完全按下时，自动对焦控制系统移动镜头将使图像清晰（关于自动对焦机理的细节请参见 1.7.1 节，在该节的后面进一步讨论移动镜头的电动机）。

被摄物体发出的光经过透镜系统聚焦在图像传感器上。这是一种典型的电荷耦合器件（CCD）（见 2.10 节），其中包含很多小的感光单元阵列，称为像素。在机电快门被打开的简短时间间隔内，像素暴露于从透镜透过的光中。落在感光单元上的光被转化成少量电荷，当曝光结束时，这些电荷在处理和存储在存储卡内之前，被读取并存储在寄存器中。

552~553

传感器不分辨颜色，所以为了拍摄彩色相片，通常在感光单元阵列前放置彩色滤波阵列。每个感光单元有三个独立的滤光片，分别为蓝色、绿色和红色。滤光片最常用的设计是采用拜耳（Bayer）阵列。这种阵列在一个模式中包含上面三种颜色，并且相同颜色的滤光片不相邻。而且绿色滤光片的数量分别是红色和蓝色的两倍，这是因为绿色大约在可见光谱的中心，并且可以提供更多的信息。在此阶段的图像是红色、绿色和蓝色像素的马赛克。因为文件还没经过任何处理，此阶段的像素文件也称为原始文件（RAW file）。为给出特定像素的完整颜色，需要采用一种算法，在这种算法中，分配给特定像素的颜色是由相邻像素颜色的强度决定的。

信号处理的下一个阶段是压缩文件使之占用尽可能少的内存，用这种方法可以在存储卡中存储比原始状态更多的文件。一般的压缩文件格式为 JPEG，这是联合摄影专家小组的简称。JPEG 采用的原理为：在许多相片中，相同区域的很多像素都是相同的，所以更有效的是只存储一个像素，其他地方则说明重复这个像素，这就避免了存储每一个像素的相同信息。

要求的曝光时间是由相机的微控制器对传感器输出的响应决定的，例如光电二极管探测光的强度。微控制器给出输出信号用于控制光圈和快门的速度。数码相机的光圈驱动系统可以用步进电动机，步进电动机根据从微控制器接收到的信号打开或关闭一组隔膜叶片。数码相机的快门机构的形式通常如图 24-19 所示。机械快门包含两组幕帘，每一组幕帘都由弹簧锁控制。当电磁铁中没有电流通过时，弹簧力使锁闩到达一个位置，这使得上面的幕帘下降并与下面的幕帘重叠。当电磁铁中流过电流时，锁闩旋转，使上幕帘提升。由于下幕帘控制弹簧锁的电磁铁通过电流，所以下幕帘最初位于于底部。当该电磁铁中的电流切断时，下幕帘将上升。因

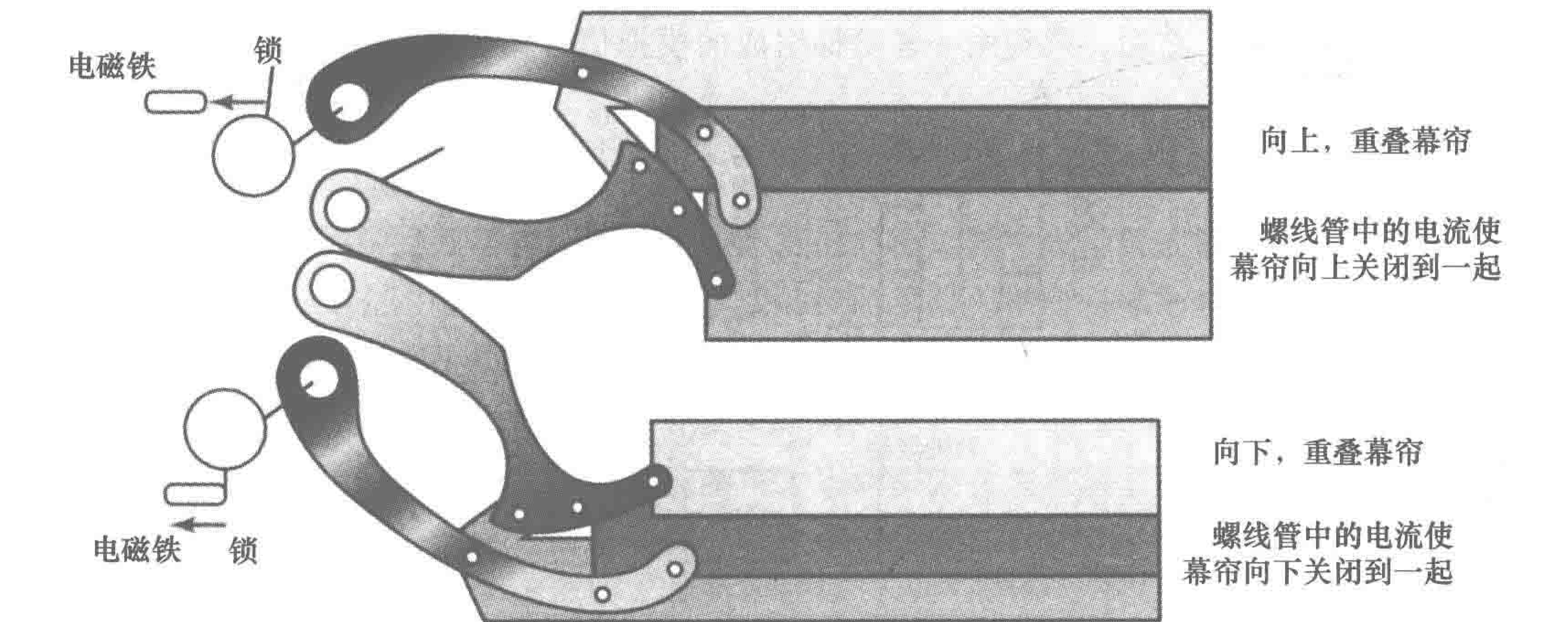


图 24-19　快门机构

554

此，向图像传感器打开光圈是由接通上磁铁中的电流和切断下磁铁中的电流之间的时间决定的。

聚焦需要一种机构来控制镜头移动。这种机构通常为一个由一系列压电元件组成的超声波电动机，比如锆钛酸铅（PZT）压电陶瓷。当电流施加到压电元件时，元件根据电流的极性扩张或收缩（如图24-20a所示）。PZT元件粘在一条薄弹性钢条的两面。当其上有电势差时，使PZT扩张或收缩，钢条弯曲（如图24-20b所示）。当相反极性的电流施加到元件上时，元件会向相反方向弯曲（如图24-20c所示）。因此，把交流电流加到一系列这样的元件组成的一个圆环上时，就形成一个位移波，位移波既可以在压电元件环中顺时针传播也可以逆时针传播。这种位移波的振幅大概只有0.001mm。在PZT元件的外面贴有小齿的条形材料，当位移波绕着PZT元件传播时，能够推动透镜支架（如图24-20d所示），从而驱动聚焦元件运动。

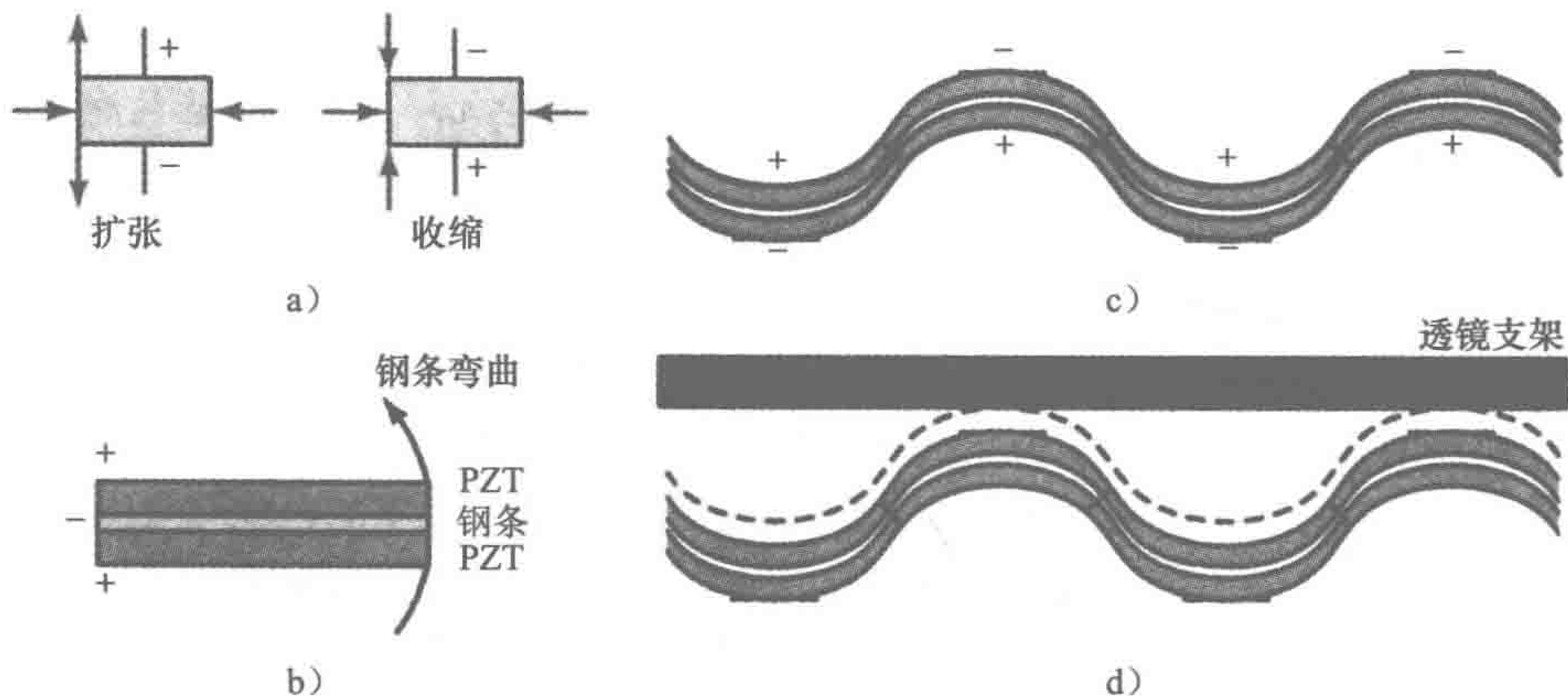


图24-20　压电电动机

为解释前面章节中讨论的建模技术，现考虑超声波电动机。电动机需产生转矩T，使电动机环转动一定的角位置θ。因为环是非常轻的，所以和环与环之间的摩擦相比，我们忽略它的惯性，假定摩擦力和角速度ω成正比，则$T=c\omega=c\mathrm{d}\theta/\mathrm{d}t$，其中$c$为摩擦常量，积分可得：

$$\theta = \frac{1}{c}\int \mathrm{d}t$$

则传递函数为$G(s)=1/cs$。

超声波电动机的控制系统如图24-21所示。其中y_n为第n个输入脉冲，x_n为第n个输出脉冲。微控制器的比例控制增益为K，控制器的输入为y_n-x_n，输出为$K(y_n-x_n)$。输出信号经过数模转换器（DAC）之后，得到由一些阶梯组成的模拟信号输出（如图24-21所示）。该电

555

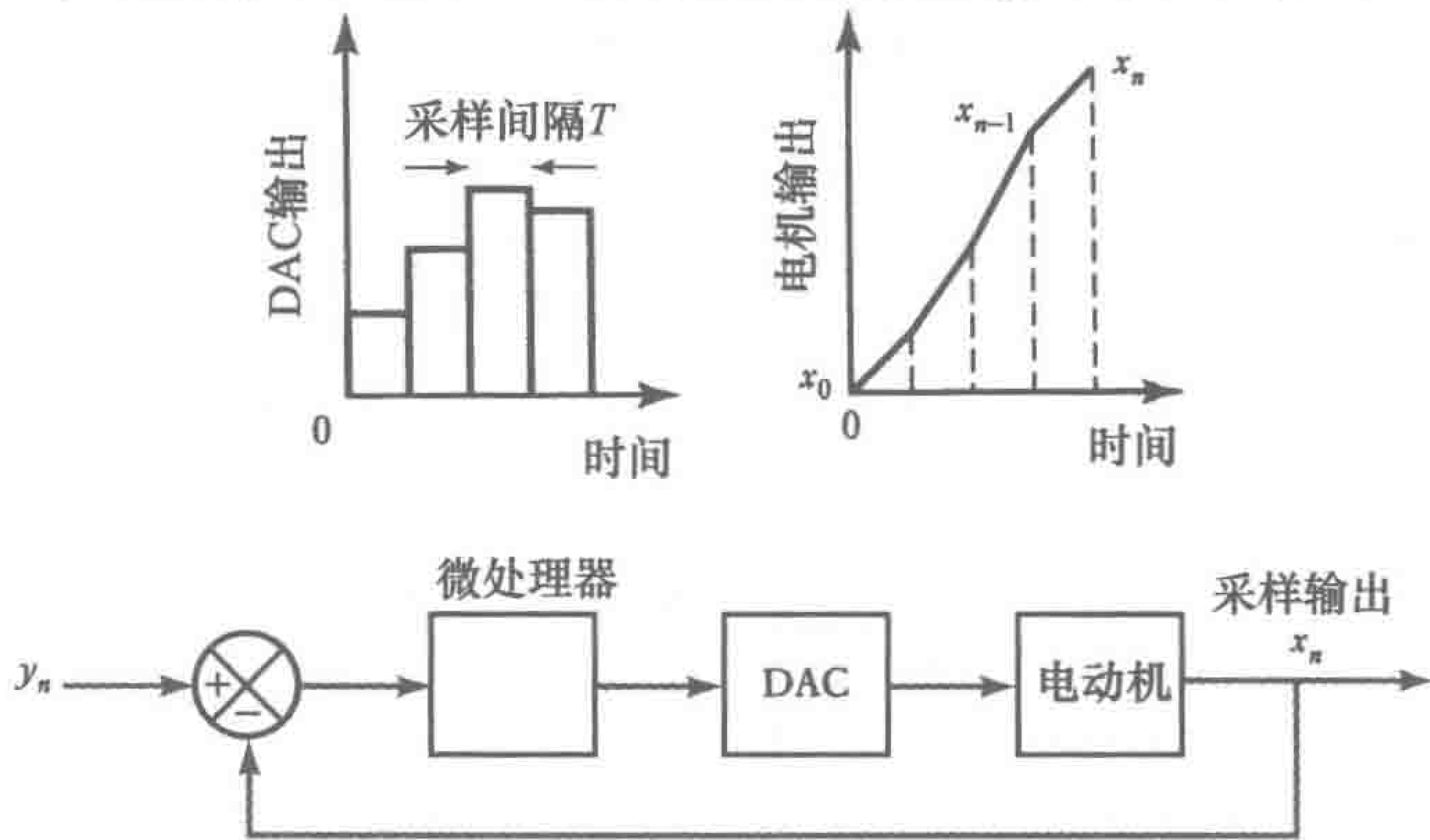


图24-21　控制系统

动机起到了积分器的作用，因此它的输出是阶梯形信号下面积的 $1/c$ 倍。每一阶梯有面积：（该阶梯的 DAC 输出改变）$\times T$，因此，

$$x_n - x_{n-1} = (x_{n-1}\text{ 次 DAC 输出})T/c = K(y_{n-1} - x_{n-1})T/c$$

因此，

$$x_n = [1 - (KT/c)]x_{n-1} + (KT/c)y_{n-1}$$

若 $\frac{K}{c}=5$，采样间隔 $T=0.1\text{s}$，则

$$x_n = 0.5y_{n-1} + 0.5x_{n-1}$$

如果聚焦控制系统的输入脉冲为 1，之前没有任何输入，即 $y_0=1$，$y_1=1$，$y_2=1$，…，则，

$$x_0 = 0$$
$$x_1 = 0.5\times 0 + 0.5\times 1 = 0.5$$
$$x_2 = 0.5\times 0.5 + 0.5\times 1 = 0.75$$
$$x_3 = 0.5\times 0.75 + 0.5\times 1 = 0.875$$
$$x_4 = 0.5\times 0.875 + 0.5\times 1 = 0.9375$$
$$x_5 = 0.5\times 0.9375 + 0.5\times 1 = 0.968\,75$$
$$x_6 = 0.5\times 0.968\,75 + 0.5\times 1 = 0.984\,375$$
$$x_7 = 0.5\times 0.984\,365 + 0.5\times 1 = 0.992\,187\,5$$

聚焦需要 7 个采样周期，即 0.7s。这个时间很长。若取值 $\frac{KT}{c}=1$，则微分方程变为 $x_n=y_{n-1}$，因此可以得到

$$x_0 = 0$$
$$x_1 = 1$$
$$x_2 = 1$$
$$x_3 = 1$$
$$\cdots$$

556

这意味着，达到所需位置只要一个采样周期。这是一个很快的响应。采样频率越高，响应速度越快。这种响应形式叫作**无差拍响应**。

24.2.4　汽车控制系统

现代汽车往往包含很多电子控制系统，这些系统主要有以下几个分类。

1. 动力总成控制系统

动力总成控制系统是发动机和传动控制系统的总称。其中发动机控制单元（EUC）的作用在于保证发动机时刻在最佳状态下运转，常用的控制手段包括燃料喷射控制、化油器控制、点火正时控制、怠速控制和抗爆震控制等。发动机舱中安装有许多传感器，发动机控制单元会从这些传感器中读取相关数据进行理解，并据此来对发动机驱动器做出相应的调节。传动控制主要指自动传动，通常会采用单个发动机控制单元来同时进行发动机控制和传动控制。一个发动机控制单元含有一个微控制器，而一些操作软件则被存储在 EPROM 存储器或闪存中。图 24-22 展示了发动机控制系统的一些基本输入和输出元件。

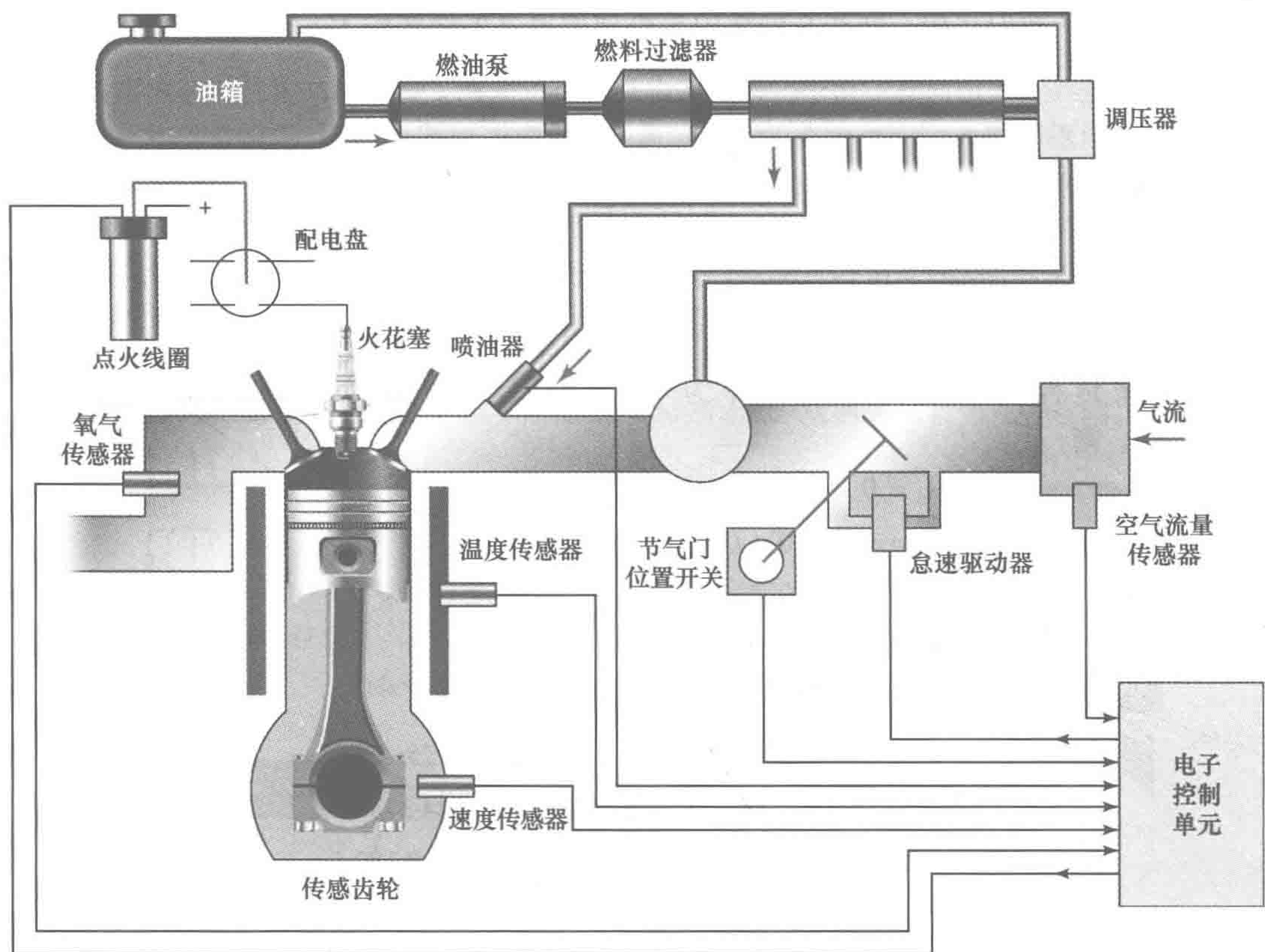


图 24-22　发动机控制系统

2. 车身控制系统

车身控制系统包括悬架控制、转向控制、巡航控制、制动控制和牵引控制等。

3. 人体控制系统

该系统包括汽车内部的空调、仪表显示、安全系统、通信系统、安全气囊和尾部障碍物探测等。

汽车中会使用一个专门的网络来完成多个控制系统之间的通信，而该网络的标准版本便是控制器局域网络（CAN）（详见 15.6.4 节）。

以下是汽车控制系统中常见控制部分的讨论。

空燃比（AFR）是指内燃发动机中空气和燃料之间的质量比。当有足够的空气支持燃料的完全燃烧时，此时的空燃比被称为化学当量比。对于汽油燃料来说，空燃混合物的化学当量比约为 14.7：1，也就是说每克燃料需要 14.7 克空气支持其完全燃烧。我们用 λ 来表示实际空燃比和所给混合物对应的化学当量比之间的比例。当 $\lambda = 1.0$ 时，表示该混合物正处于化学当量状态；当 $\lambda < 1.0$ 时，混合物处于较浓状态；当 $\lambda > 1.0$ 时，混合物则较为稀薄。正是因为如此，排气氧传感器（EGO）是发动机燃料反馈闭环控制中的关键传感器，其产生的电压输出与排出尾气的氧气量相关。其中最常见的是二氧化锆排气氧传感器。当环境温度较高时，二氧化锆会对比排出尾气中氧气的含量和外界环境中的氧气含量，并据此产生一个电压。若混合物正好符合化学当量比，该传感器给出的输出电压约为 0.45V；其反馈电压的变化范围在 0.2 ~ 0.8V，当反馈电压为 0.2V 时，代表该混合物比较稀薄，反之若电压达到 0.8V，则说明混合物较为浓稠（详见图 24-23）。

使用λ传感器（即排气氧传感器）的控制系统大多是针对一阶发动机的 PI 控制器。图 24-24 展示了一个基本的系统模型。发动机可以被认为是一个一阶系统（详见 20.2 节），故其传递函数有以下形式：

$$G_e(s) = \frac{K}{\tau s + 1}$$

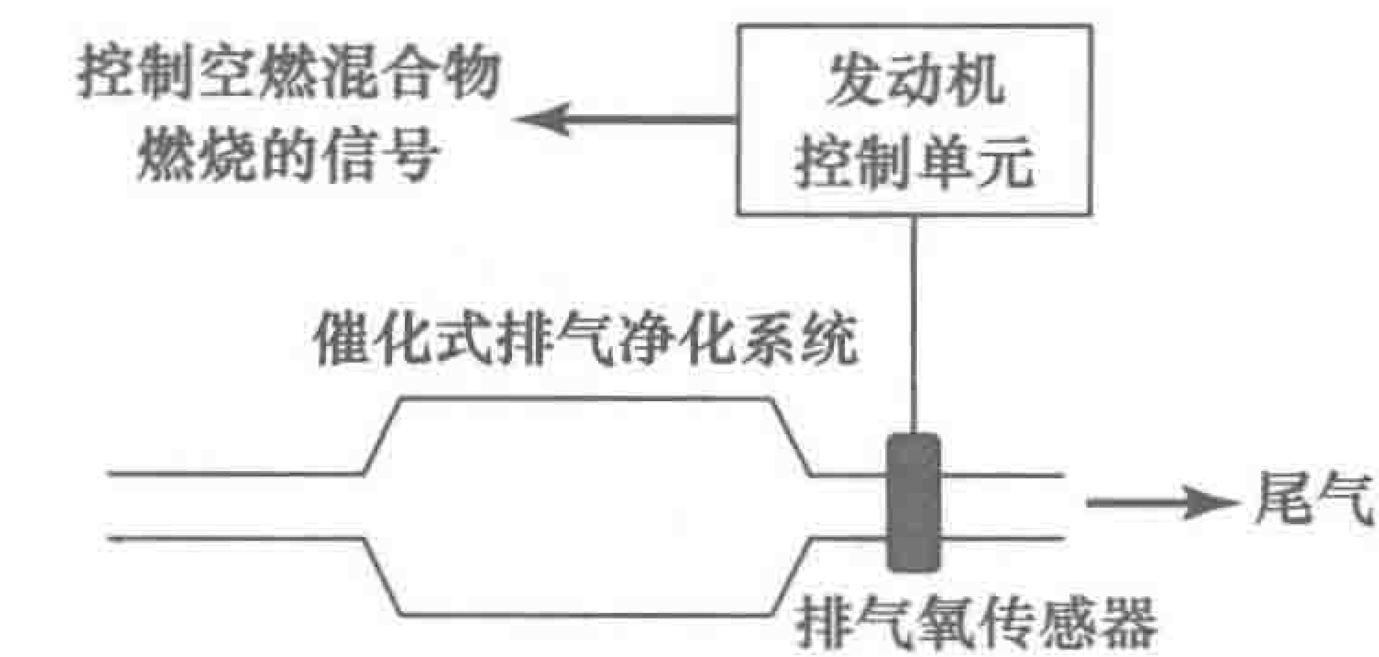


图 24-23　λ控制电路

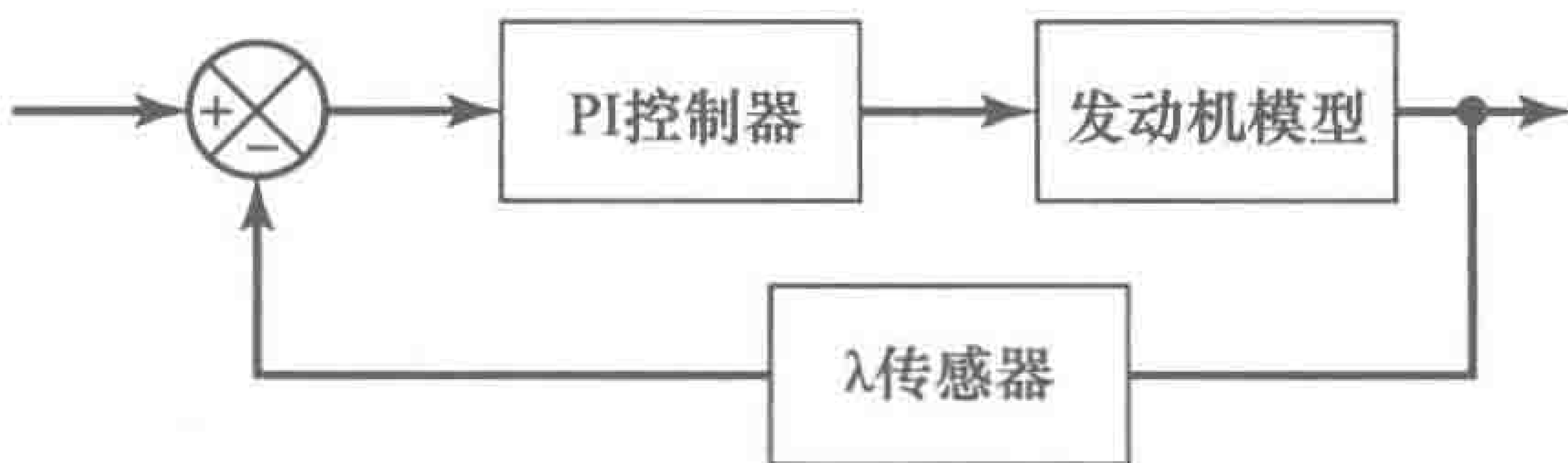


图 24-24　λ控制器系统

PI 控制器的传递函数 G_c 为 $K_P + K_I/s$，其中 K_P 是比例控制常数，而 K_I 是积分控制常数（详见 15.6.1 节）。当λ传感器的传递函数取 1 时，该闭环回路控制系统的总传递函数 $T(s)$ 为：

$$T(s) = \frac{G_e(s)G_c(s)}{1 + G_e(s)G_c(s)}$$

该传递函数给出了一个含有自然频率和阻尼比的二阶系统。于是，我们通过考虑系统的上升时间和稳定时间等因素，便可以在给定输入的条件下判断系统的状态和性能（详见 19.5 节）。

然而需要注意的是，λ传感器对氧气变化进行反馈时存在一定的延迟，该延迟时间 T_L 通常在 50～500ms，所以在考虑该延迟的情况下，我们可以在现有传递函数中加入一个延迟项，故：

$$G_e(s) = \frac{Ke^{-sT_L}}{\tau s + 1}$$

另一个汽车控制系统的例子是防抱死制动系统（ABS），防抱死制动系统的主要元件包括两个交替测量车轮转速的测速器（详见图 24-25）。若前一次测得的车轮转速值大于后一次测试值且超过某一预设值时，系统便会认为车辆发生了打滑，在此条件下，防抱死制动系统会产生一个电信号来充分地降低液压从而消除锁定制动。车轮转速传感器通常会使用变磁阻测速发电机（详见 2.4.2 节），而该发电机又包含铁齿轮。当铁齿轮转动时，会在耦合线圈中产生一系列脉冲信号，我们正是通过这些脉冲信号来测出车轮转速的。之所以选用磁性传感器，是因为车辆在使用过程中，车轮会不可避免地沾染一些泥巴、污水等污渍，而磁性传感器不同于光学传感器系统，不会受到这些因素的影响。

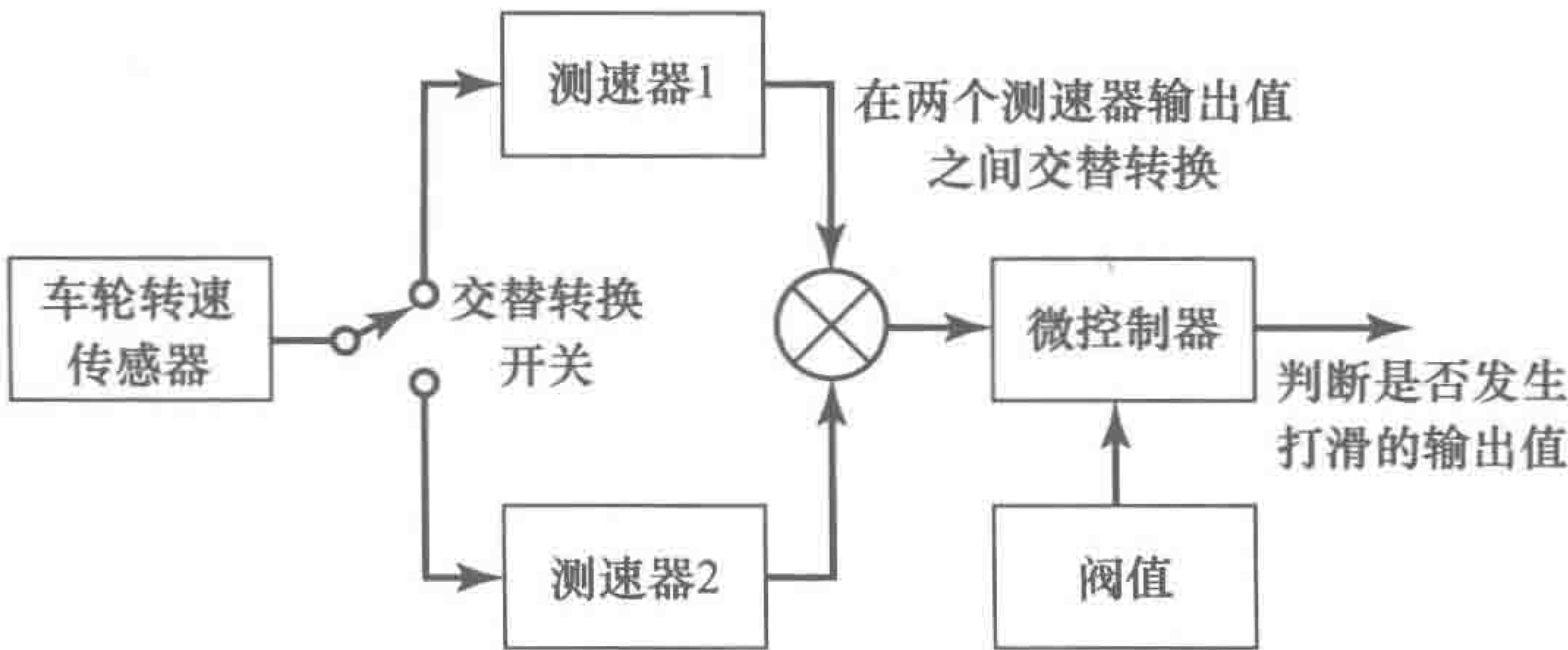


图 24-25　防抱死制动系统

24.2.5 条形码阅读器

在超市的结账口处，一个比较熟悉的场景是：购买的商品被移到一束光前面或者一个手持扫描笔移到商品的上面，这些都是为了阅读条形码，从而自动确定商品的种类和价格。条形码包括一系列宽度不同的黑白条。例如，在本书的封底有一个这样的条形码。

559

图 24-26 给出了用于零售业的条形码的基本形式。条形码代表一系列的数字。条形码的前缀代表所使用的编码方案。美国的通用产品代码（UPC）前缀有一位数字，而欧洲的商品编号（EAN）则是两位数字。UPC 码中，杂货以 0 为前缀，而药物则是以 3 为前缀。EAN 码的前缀从 00 ~ 99，因此 UPC 码也可以在 EAN 码中读出来。前缀之后是代表制造商的 5 位数字，每个制造商都有分配给自己特有数字。这就引出了代码模式的中心，并以两条比较高的线条标识。这之后的 5 位数字代表产品。最后一位数字是一个校验数，用来检验代码已经被正确读取。条形码的起始符和终止符是两条较高的线，构成了条形码的框架。



图 24-26　条形码

每个条形码被编码为 7 个 0 或 1 的数字。为了确定扫描器的扫描方向，分布在中心线两边的编码是不同的。向右边的字符有偶数个 1，所以是偶校验。对于 UPC 码，向左边的字符有奇数个 1，所以是奇校验。EAN 左边编码是一种混合形式。表 24-2 给出了 UPC 和 EAN 两种编码，UPC 用 leftA 编码，而 EAN 用 left A 和 left B 两种字符编码。

表 24-2　UPC 和 EAN 编码

十进制数	Left A 字符	Left B 字符	Right 字符
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

每个数字 1 用黑色线条表示，因此中心线右边的字符 2 用 1101100 表示。因为两条黑色线条相邻，它显示出来的是两条黑色线条的宽度，其后是一条相对较窄的空白线条，然后又是两倍宽的黑色线条，最后是两倍线条宽的空白线条，如图 24-27所示。起始符和终止符的编码均为 101，中间分隔符的编码是 01010。

560

图 24-26 所示是本书第 1 版的条形码。它使用的是 EAN 码，前缀 97 表示它是出版物，80582 代表出版商，25634 表示特定的书，最后校验数为 7。需要注意的是，条形码包括 ISBN 号码的相关部分，这个号码也是用来识别有关的出版商和书的类型。

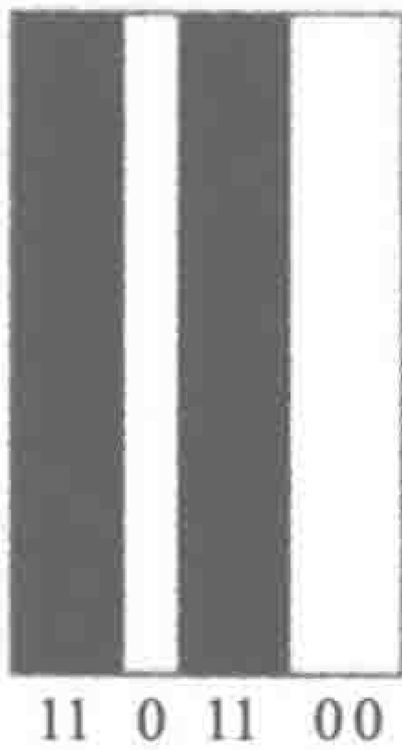


图 24-27　2 的条形码

使用校验码数字的步骤为：

1）从左边开始，计算奇数位数字（不包括校验数字）的和，如第 1 位、第 3 位、第 5 位等，然后再乘以 3。

2）从左边开始，计算偶数位数字的和。

3）把步骤 1 和步骤 2 所得的结果加起来。上述结果加上一个最小

的数之后是 10 的倍数，则这个最小的数称为校验位。

用该书的条形码 9780582256347 来说明校验数字的使用，奇数位数字之和是：9 + 8 + 5 + 2 + 5 + 3 = 32，乘以 3 得 96；偶数位数字之和是：7 + 0 + 8 + 2 + 6 + 4 = 27。两者之和是 123，所以校验数字应该是 7（123 + 7 = 130，130 是 10 的倍数）。

条形码读取包括确定黑条和白条的宽度。这可能需要使用一个固态激光器，它可以产生朝向条形码的一定强度、宽窄的光束，并用光电元件检测反射光。通常，超市版本的扫描器是形式固定的，可以用旋转镜引导光照射条形码，所以能够扫描所有的线条。为了当扫描到黑色线条时产生高电平输出，即 1；扫描到白色线条时产生低电平输出，即 0，信号调理用运算放大器把光电元件的输出信号放大，然后再用运算放大电路作为比较器。这样，一系列的 1 和 0 的结果就可以输入到连接摩托罗拉 6800 微处理器的外部接口适配器 PIA。微处理器程序的形式如下：

1）清空各种存储器地址进行初始化。

2）从输入重新恢复数据。这需要不断地检测输入，从而决定输入是 1 还是 0。

3）处理这些数据将它们转换成二进制形式的字符。输入是不同间隔的 0 或 1 组成的串行信号，间隔取决于黑色线条的宽度。微处理器系统通过编程用扫描时间除以“模块”的数量来获得“一个模块”的时间宽度，其中“一个模块”是指代表一个 0 或 1 的一个亮条或者一个暗条。程序能够决定一个暗条或亮条是代表一位数还是一位以上的数，这样就可以翻译扫描信号了。

4）处理二进制结果，显示被购物品及价格。

24.2.6　硬盘驱动器

图 24-28a 展示了一个硬盘驱动器的基本形式，它包括涂有金属涂层的磁盘，其中金属涂层可以被磁化。写/读磁头与磁盘表面的距离非常小，大约 0.1μm。数据以一系列位单元的形式存储在金属涂层中（参见 6.3.1 节）。电动机通常以 3600、5400 或 7200rev/min 的转速转动磁盘，并且需对传动臂进行定位，以便使相关的同心磁道和磁道的相关部分处于臂末端的读/

561

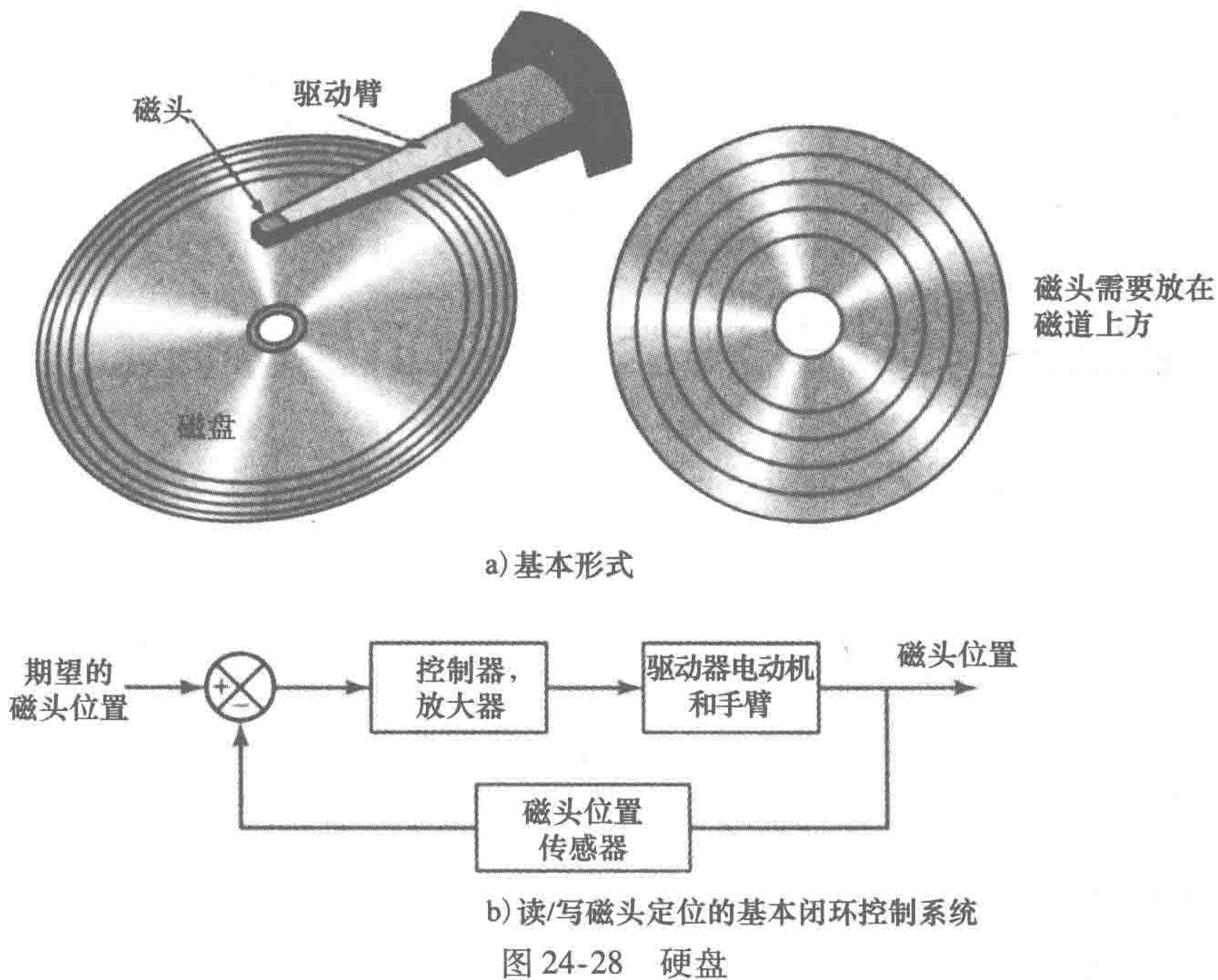


图 24-28　硬盘

写头下。为了定位磁头，由一个闭环系统控制碰头（见图 24-28b）。控制信息在格式化的过程中被写在磁盘上，这样可以区别每个磁道和磁道的扇区。所以这个控制过程就包括磁头利用这些信息达到磁盘上要求的部分。

驱动器的运动通常由音圈电动机（如图 24-29 所示）驱动传动臂旋转。音圈电动机实质上是一个安装在铁心上的线圈，当电流通过线圈时磁盘就会转动，线圈布置与动圈式扬声器一样，所以音圈电动机能够移动传动臂把磁头定位在需要的磁道上。磁头读取磁盘上的磁场，并为控制放大器提供反馈信号。

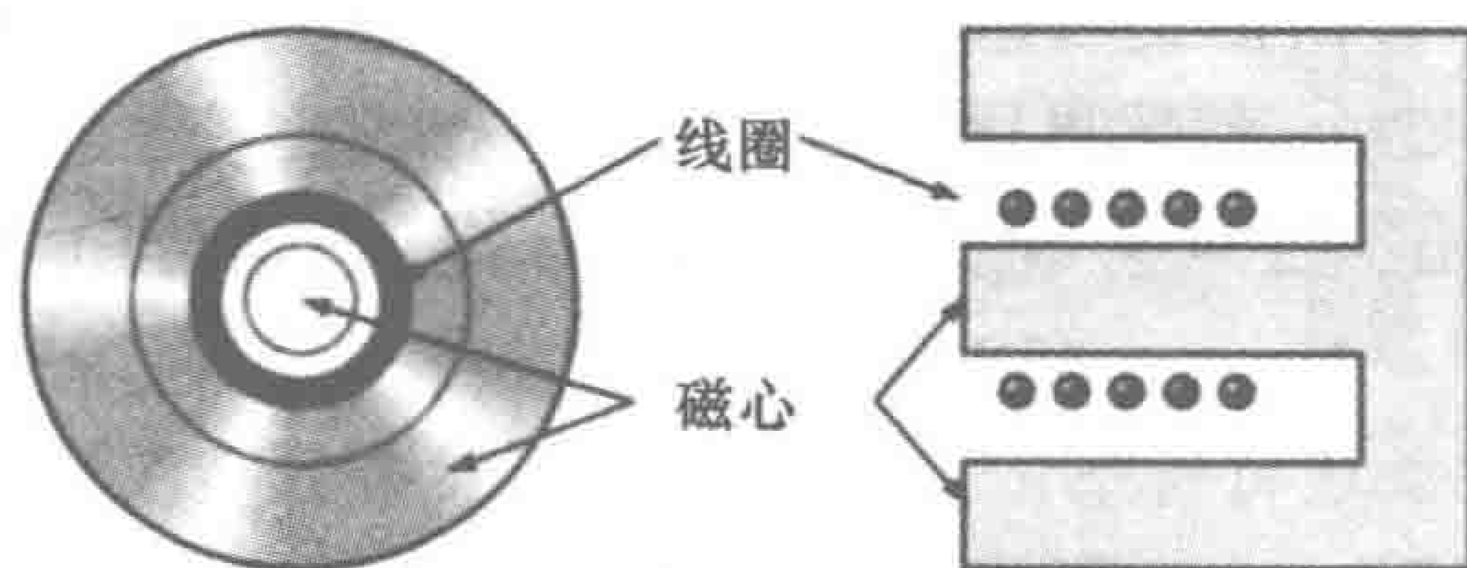


图 24-29 音圈电动机

音圈电动机是一种磁场控制的永磁直流电动机，其有相同形式的传递函数（参见 20.5 节）。因为我们考虑的是关于位移和时间的传递函数，即 20.5 节给出的速度时间函数的积分，音圈电动机有如下形式的传递函数：

$$G(s) = \frac{k}{s(Ls+R)(Is+c)} = \frac{k/Rc}{s(\tau_L s+1)(\tau s+1)}$$

其中（$\tau s+1$）项通常情况下约等于 1，所以传递函数近似为

$$G(s) = \frac{k/Rc}{s(\tau_L s+1)}$$

因此，图 24-28a 所示的闭环控制系统，控制放大器有比例增益 K_a，磁头位置传递增益 1。可得总的闭环传递函数为

$$\frac{X(s)}{R(s)} = \frac{K_a G(s)}{1+K_a G(S)}$$

给出了输出信号 $X(s)$ 和要求的输入信号$R(s)$之间的关系。

因此若 $G(s)=0.25/s(0.05s+1)=5/s(s+20)$，且 $K_a=40$，则有

$$X(s) = \frac{200}{s^2+20s+200}R(s)$$

对于单位阶跃输入，即 $R(s)=1/s$，输出为

$$X(s) = \frac{200}{s(s^2+20s+200)}$$

二次项也是 $s^2+2\zeta\omega_n s+\omega_n^2$ 这种形式（见 20.3.1 节），所以固有频率 $\omega_n=\sqrt{200}$，阻尼因子是 $\zeta=\frac{10}{\sqrt{200}}$。因此，我们能够得出该二阶系统对于阶跃输入信号的响应和该系统达到稳定的时间。例如，在稳态值的 2% 内需要的建立时间是（参见 19.5 节）是$\frac{4}{\zeta\omega_n}$，所以 $4/10=0.4\text{s}$。这是比较长的时间，所以我们需要考虑怎样才能减少到毫秒级。我们可考虑用 PD 控制代替有比例增益的放大器。

24.3 机器人学

机器人学（robotics）是指与机器人的设计、制造和应用相关的技术学科。**机器人**是一种可

以描述为“智能机器”的设备；而“非智能机器”则通常是指需要人工操作来控制运转的设备。智能机器将设备操控者从人变成了电脑，不借助于操作人员对其下一步行为做出决策，而依靠其自身对环境信号的响应实现自动运转。

562~563

以下是在机器人学发展史中的一些关键性历史事件。

1922 年，捷克作家卡雷尔·卡佩克（Karel Capek）在他的戏剧《通用机器人》（Universal Robots）中首次提出了“机器人”这一术语。该戏剧讲述了一个制造智能人（被称为“robots”）的工厂，而“robot”一词源自于斯拉夫语中的 robota，意为劳动力。

1942 年，科学幻想小说作家伊萨克·阿西莫夫（Isaac Asimov）在《说谎者》（Liar）中引入了机器人学的术语，然后提出了机器人学的三大法则。

1956 年，乔治·德沃尔（George Devol）和约瑟·英格伯格（Joe Engelberger）成立了一家名为 Unimation 的公司，并研发出了第一款工业机械臂，取名为 Unimate。

1961 年，乔治·德沃尔作为 Unimate 机器人基础的“Programmed Article Transfer”获得了美国专利，第一代工业机器人就是由它们安装的。通用汽车公司首次在生产线中安装了 Unimation 的机器人，用于排列和堆叠压铸金属热片。

1967 年，通用汽车公司安装了第一台点焊机器人，使得 90% 的汽车外壳焊接操作实现自动化。

1973 年，库卡机器人公司（KUKA）研发出了第一台有 6 个机电驱动轴的机器人 Famulus。

1974 年，瑞典机械工程公司 ASEA 研发了第一款微处理器控制的工业机器人，其机械臂模仿了人类的手臂运动，采用了英特尔 8 位微处理器。

1978 年，Unimation 公司推出一款可编程通用装配机（PUMA），应用于通用汽车公司的小部件组装线。

1979 年，日本的 Nachi 公司首次研制出电机驱动的点焊机器人，替代传统的液压驱动机器人。

1986 年，本田公司（Honda）推出了第一款人形机器人。

1987 年，国际机器人协会成立。

24.3.1 机器人学的三大法则

以下机器人学的三大法则是由科幻小说作家伊萨克·阿西莫夫于 1942 年提出的。

1）机器人不得伤害人类个体，或者目睹人类个体将遭受危险而袖手旁观。

2）机器人必须服从人给予它的命令，当该命令与第一法则冲突时例外。

3）机器人在不违反第一、第二法则的情况下要尽可能保护自己的生存。

机器人并非生来便懂得这三大法则，要使它们遵守这些法则，机器人的创造者必须选择在编程设计中编入这几条法则。到目前为止，机器人的智能程度还不足以让它们获悉自己何时会造成痛苦或伤害并及时停止。尽管如此，我们仍然可以在机器人的制造过程中为它们加入一些物理防护措施以防止事故的发生，例如缓冲器、警报器、安全护笼等。

564

阿西莫夫后来还加入了凌驾于其他法则之上的第四条法则，也称为“零法则”：

机器人不得伤害人类，或者对人类即将受到的伤害不作为。

24.3.2 机器人组件

以下罗列了一些机器人所包含的基本组件。

1）**操作臂**。操作臂主要由连杆、关节及其他结构元件构成，是机器人的主体部分，负责实施操作。

2）**末端执行器**。末端执行器是机器人的“手”，它与操作臂的最后一个关节相连，用于完成对具体对象的处理或与相关的机器相连。夹钳是一种常见的末端执行器，它可以通过两个

“手指”的开合来实现对物件的抓取和放置。抓取物件的过程可以通过钳子与物件的摩擦力或环绕式狭口来实现。而对于大型物体（如汽车挡风玻璃）的抓取则通常采用真空夹持器来完成。这种情况下，为了保证足够的吸力，物体需要有一个平滑的表面。不过机器人末端执行器不仅仅代表了夹持器，如机器人工具变换器、机器人喷漆枪、机器人倒角工具、机器人电弧焊接枪等设备也是末端执行器的一部分。

3）**作动器**。作动器的作用类似于人类的肌肉，用于移动机器人的关节和连杆。最常见的作动器主要有两种，即电动机和线性作动器。前者主要用于转动轮子或齿轮，而后者由压缩空气或油驱动活塞在缸内运动。

4）**传感器**。传感器被机器人用于采集环境信息和自身连杆、关节等状态信息，它的设计仿照了人类手指的机械性能和触觉感受器。举例来说，旋转式电位器可以用来监测手部关节之间的角度变化；触力传感器则可以探测手指与手掌的状态，从而判断物体与手之间是否存在接触，两者之间的压强是多少等。图 24-30 展示了一个传感器阵列，图中显示的柱塞阻断了 LED 和传感器之间的光束。除此之外，我们也可以通过测量两点之间的导电弹性体（如含碳橡胶）565的阻值来达到相同的目的，阻值会随着所施加的力的大小而变化（如图 24-31 所示）。这种传感器主要使用了由弹性线构成的网格状结构，而我们需要测量的是网格交叉点处的阻值。机器人上所装备的每一个触觉传感器会根据其驱动状态的不同探测到关于物体形状和大小的不同信息。在机器人的拇指上安装一个触觉 - 滑觉传感器可以探测到抓取的物体是否发生滑动，并因此增加抓取所施加的力；这可以通过解释触觉传感器阵列的输出结果或采用一个特别设计的滑觉传感器来实现。

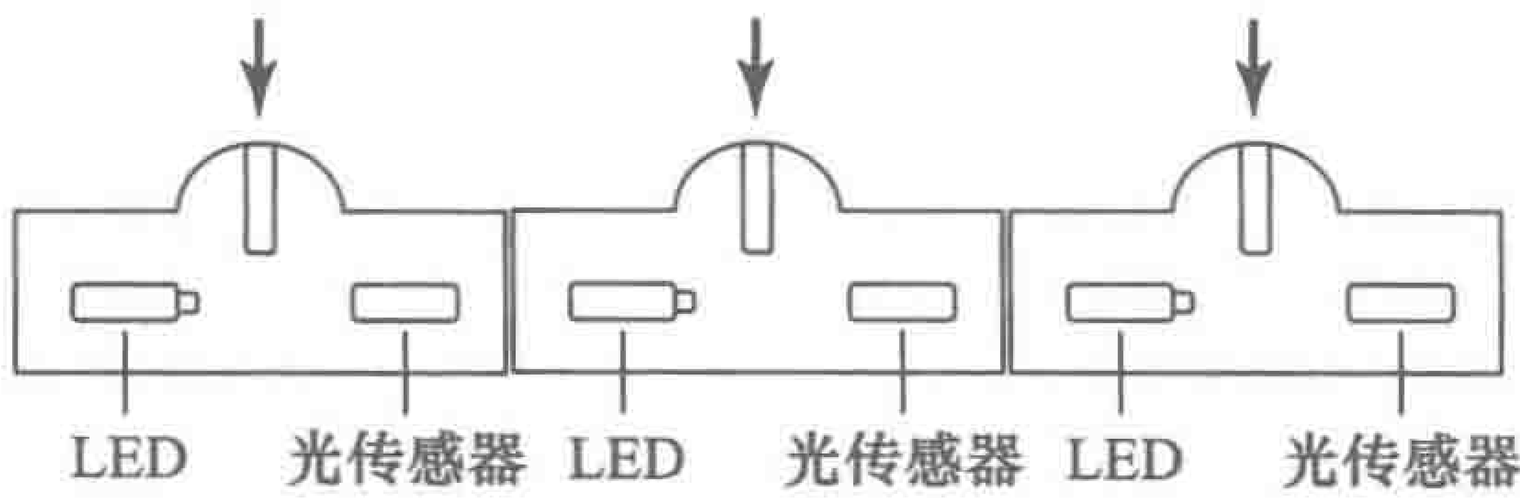


图 24-30　触觉传感器的一个示例。接触会引发柱塞阻断光束，从而使传感器探测到该信号

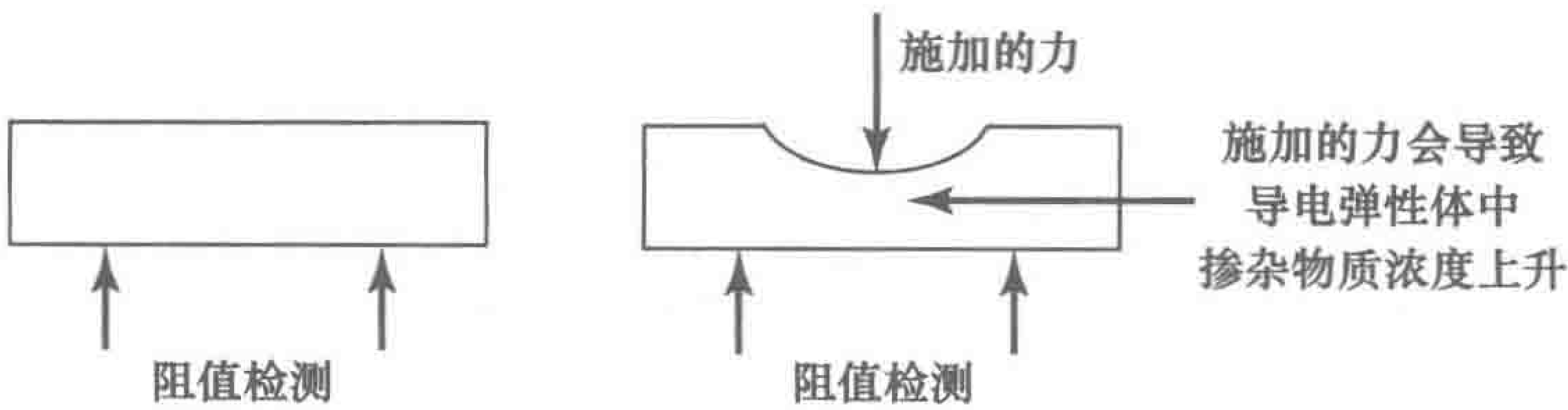


图 24-31　导电弹性体的阻值会随着接触点压力的变化而改变

5）**控制器**。为了使机器人可以执行不同的任务，我们需要控制机器人的机械结构。对机器人的控制过程主要包括：对传感器信息的提取和处理，将该信息转化为对作动器的命令，以及控制机器人的行为。而控制器背后的“大脑”则是计算机。

24.3.3　机器人应用

以下是一些机器人的具体应用。

1）**机器人上下料**。机器人常用于为其他机器提供零部件或从其他机器上移除已经过加工的零部件。

2）**取放操作**。机器人可拾取零件，并将零件放置于托盘上；也可用于拾取两个零件并进行装配，例如把零件放入一个盒子中，把零件放入炉中，或把零件从炉中取出。

3）**焊接**。机器人末端执行器可以是一把焊枪，用于零件之间的焊接。

4）**喷漆**。机器人末端执行器可以是一把喷漆枪，用于组装件或零件的喷漆作业。

5）**组装**。机器人可用于不同零件的定位和识别，并完成零件之间的组装工作。

6）**检查**。机器人可使用 X 光、超声波或“视觉”等对部件进行检查。

7）**助残**。机器人可以通过人工智能肢体来协助残障人士完成日常生活任务。

8）**危险区域及难达区域操作**。机器人能被设计在人类难以抵达的环境中进行操作。

24.3.4 Arduino 机器人

Arduino 网站描述了一款使用 Arduino 控制板和电机板的基本轮式机器人（详见 10.3.4 节）。这款机器人的控制板和电机板均配有微控制器。包括机器人的硬件、软件和资料文档在内的所有组件都是免费且开源的，机器人可以通过 Arduino 对其进行编程。Arduino 机器人所采用的处理器都预装有 bootloader，用户可以通过该程序向机器人上传新代码而不需要借助于外部硬件编程器。以下这段小程序可以使机器人重复地前后移动。当程序上传后，出于安全考虑，请拔除机器人的 USB 连接线，因为当机器人通过 USB 连接时，其电动机处于脱离状态。当打开电源时，机器人便会开始运动。 566

```
#include <ArduinoRobot.h> // 导入机器人库
void setup()
{
Robot.begin(); // 初始化库
}
void loop()
{
    Robot.motorsWrite(255,255);// 设置电机的速度为全速
    delay(1000);// 运动 1s
    Robot.motorsWrite(0,0); // 设置停止运动
    delay(1000);// 停止运动 1s
    Robot.motorsWrite(-255,-255);// 设置电机反转
    delay(1000); // 反转运动 1s
    Robot.motorsWrite(0,0); // 设置停止运动
    delay(1000);// 停止运动 1s
}
```

用户也可以使用红外传感器对机器人实现更复杂的程序控制，使得机器人实现左右、向前等运动，从而使其在不撞墙的前提下顺利走出迷宫。

本章小结

机械电子学将机械工程、电子学、计算机智能控制集成于产品设计和制造中，并协调同步发展。它寻求一种多学科集成的解决方法，而不是用单独的学科方法。在研究解决方案时，需要建立模型，以便预测该方案是如何工作的。

习题

24.1 提出下列问题可能的大概设计方案：

(a) 炉子的温度控制器。

(b) 设计一个把沿传送带运动的物品按大、中、小分类的机构，能够使物品按大小不同分到不同的滑槽以便分装。

(c) 设计一个 $x-y$ 绘图仪（该机器绘制图形以表明当输入 y 变化时，x 是如何改变的）。

研究作业

24.2　研究汽车上应用的防抱死制动系统，并描述它的工作原理。

24.3　研究点阵打印机中使用的机构，并描述它的工作原理。

24.4　研究车用 CAN 总线协议。

设计作业

24.5　设计一个数字温度计系统，显示温度范围为 0 ~ 99℃，你可以考虑采用带有 RAM 和 ROM 芯片的微处理器或者微控制器的方案。

24.6　设计一个数字欧姆表，将显示连接到两表笔之间电阻的阻值。你可以考虑基于单稳态多谐振荡器的方案，如 74121，它将提供一个与连接电路的时间常量 RC 相关的宽度的脉冲。

24.7　设计一个能够显示大气压力的数字气压计，你可能采取的方案是基于 MPX2100AP 压力传感器。

567 ~ 568

24.8　设计一个用来控制直流电动机速度的系统。你可以考虑采用 M68HC11 评估板。

24.9　设计一个包含 PLC 的系统使得每次在传送带上放 4 个箱子。

第七部分　附　录

附录 A　拉普拉斯变换

A.1　拉普拉斯变换

考虑一个关于时间的函数量。我们可说这个量处于**时域**中，用函数$f(t)$表示。在很多问题的研究中，我们仅考虑时间大于等于 0 的情况，即 $t \geqslant 0$。为了得到该函数的拉普拉斯变换，我们将该函数乘以 e^{-st}，然后对其从时间 0 到无穷进行积分。在积分中 s 是一个常量，单位为时间的倒数。积分的结果称为拉普拉斯变换，所得的方程称为 s **域**方程。因此，对于一个时域函数$f(t)$，其拉普拉斯变换记为$\mathcal{L}\{f(t)\}$，由下式给出：

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t)\,dt$$

该变换所处的时域是**单边的**，即 $0 \sim +\infty$，而非 $-\infty \sim +\infty$ 的全时域。

对于一个 s 域中的量，我们可以对其进行各种代数运算，如加、减、乘、除等。我们以通常的方式，能处理任何代数量。对于一个原函数，假如其为时域中的微分方程，难以进行代数运算，则可以通过拉普拉斯变换将时域变换到 s 域，得到相对简单的表达式。要获得方程在时域中的变化结果则需要进行反变换，即找到 s 域的表达式在时域中对应的函数。

在 s 域中，函数可以写成 $F(s)$，即关于 s 的函数。通常拉普拉斯变换函数用大写 F 表示，而时变函数用小写 f 表示。因此：

$$\mathcal{L}\{f(t)\} = F(s)$$

对于反变换，由拉普拉斯变换得到的时域函数记为：

$$f(t) = \mathcal{L}^{-1}\{F(s)\}$$

表示 $f(t)$ 是拉普拉斯变换函数 $F(s)$ 的反变换。

A.1.1　根据第一个法则的拉普拉斯变换

为了阐明一个时域的量到 s 域的变换，考虑一个函数，在时间大于或者等于 0 时，其值为 1，即 $f(t)=1$，$t \geqslant 0$，这是一个单位阶跃函数。如图 A-1 所示。

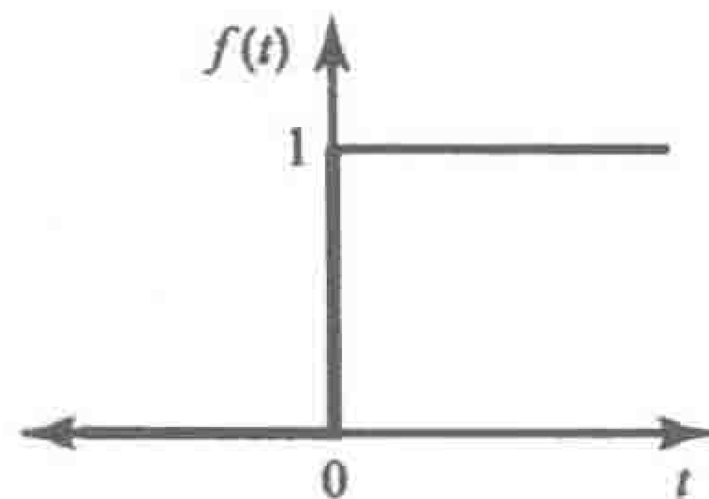


图 A-1　单位阶跃函数

拉普拉斯变换为

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^{\infty} 1 e^{-st} dt = -\frac{1}{s}\left[e^{-st}\right]_0^{\infty}$$

$t=\infty$ 时，$e^{-st}=0$；$t=0$ 时，$e^{-st}=1$，则有

$$F(s) = \frac{1}{s}$$

作为另一个例子，下面给出了根据第一法则的函数 e^{at}（其中 a 为常量）的拉普拉斯变换。对函数 $f(t)=e^{at}$，有

$$F(s) = \int_0^\infty e^{at}e^{-st}dt = \int_0^\infty e^{-(s-a)t}dt = -\frac{1}{s-a}[e^{-(s-a)t}]_0^\infty$$

当 $t=\infty$ 时，方括号内量的值为0；当 $t=0$ 时，方括号内量的值为1。故

$$F(s) = \frac{1}{s-a}$$

A.2　单位阶跃函数和脉冲函数

系统常用的输入函数有单位阶跃函数和脉冲函数。下面给出了这两种函数拉普拉斯变换的推导过程。

A.2.1　单位阶跃函数

图 A-1 为单位阶跃函数的曲线。当 $t=0$ 时产生阶跃，有以下方程：

$$f(t) = 1 \quad t > 0$$
$$f(t) = 0 \quad t < 0$$

阶跃函数描述了一种函数值从0到某一稳定值的突变，例如，当电路突然接通时，其电压的变化。

鉴于上述原因，单位阶跃函数仅用 $f(t)=1$ 表示是不够的，因为这有可能表示不论变量 t 为正值还是负值，函数值恒为1。单位阶跃函数在 $t=0$ 时从0切换到 +1，按惯例可用符号 $u(t)$ 或者 $H(t)$ 表示，H 源于 O. Heaviside，因此它有时候也称为**海维塞德**（Heaviside）**函数**。

如前一节所推导的，单位阶跃函数的拉普拉斯变换为：

572

$$F(s) = \frac{1}{s}$$

则对于高度为 a 的阶跃函数，其拉普拉斯变换为

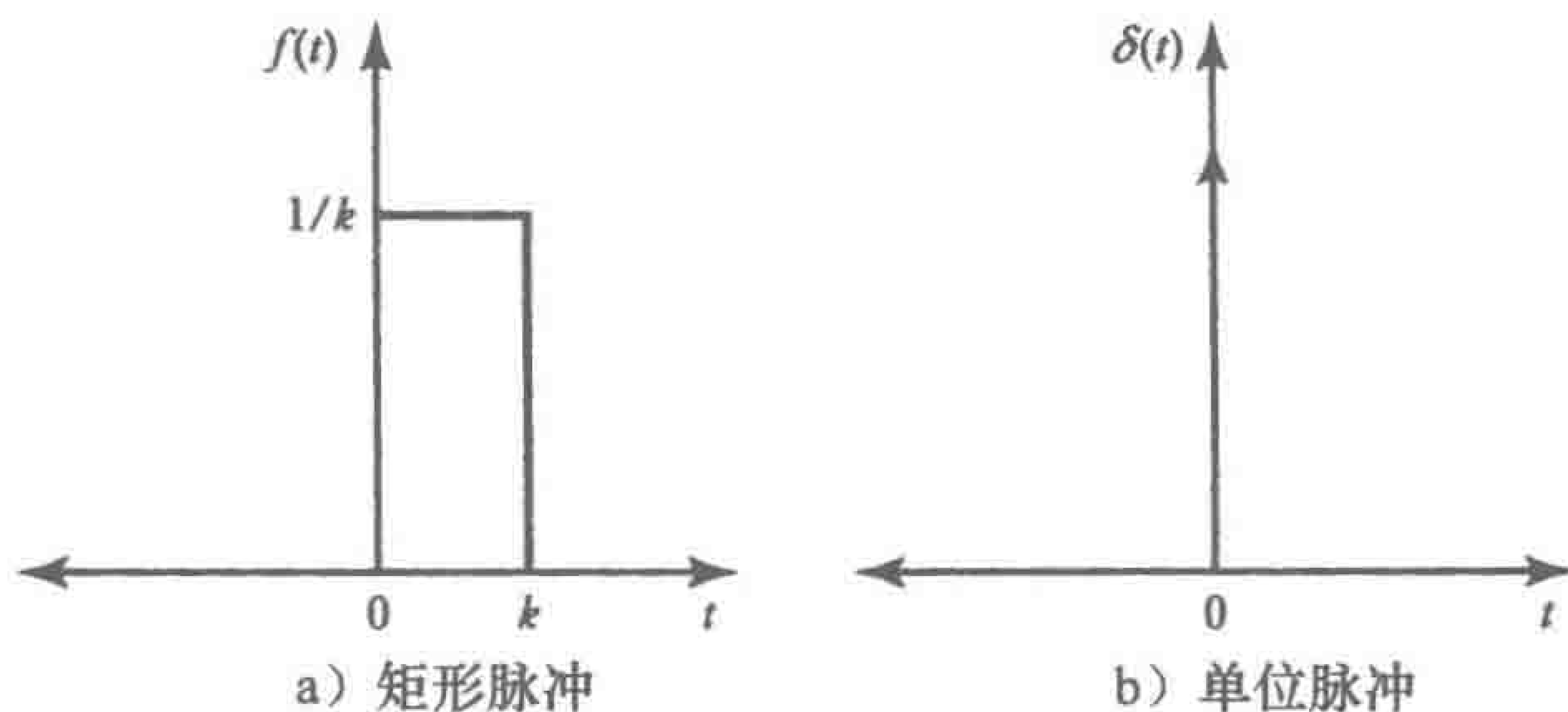
$$F(s) = \frac{a}{s}$$

A.2.2　脉冲函数

假设在时间 $t=0$ 时，有一个高度为 $1/k$，宽度为 k 的矩形脉冲，即脉冲图形的面积为1。脉冲图形如图 A-2a 所示。该脉冲可由下式描述：

$$f(t) = \frac{1}{k} \quad 0 \leqslant t < k$$

$$f(t) = 0 \quad t > k$$



a）矩形脉冲　　b）单位脉冲

图 A-2　矩形脉冲与单位脉冲

如果在保持脉冲区域面积为1的前提下，缩小脉冲宽度（即减小 k 的值），则脉冲的高度增大。因此，当取到极限 $k\to0$ 时，可以在 $t=0$ 的时刻得到一条垂直的脉冲曲线，其高度为无穷大。其图像为一个单点处的尖峰，除此点外值都为0（如图 A-2b 所示）。该图像代表一个脉

冲，又由于其包络的面积为 1，因此为单位脉冲。该函数可用 $\delta(t)$ 表示，又称为**单位脉冲函数或狄拉克德尔塔**（Dirac delta）**函数**。

图 A-2a 中描述的单位矩形脉冲函数的拉普拉斯变换由下式给出：

$$F(s) = \int_0^\infty f(t)\mathrm{e}^{-st}\mathrm{d}t = \int_0^k \frac{1}{k}\mathrm{e}^{-st}\mathrm{d}t + \int_k^\infty 0\mathrm{e}^{-st}\mathrm{d}t = \left[-\frac{1}{sk}\mathrm{e}^{-st}\right]_0^k = -\frac{1}{sk}(\mathrm{e}^{-sk} - 1)$$

为得到单位脉冲函数的拉普拉斯变换，我们需要得到当 $k \to 0$ 时上式的值，将其中的指数项作为级数展开，可得：

$$\mathrm{e}^{-sk} = 1 - sk + \frac{(-sk)^2}{2!} + \frac{(-sk)^3}{3!} + \cdots$$

因此有：

$$F(s) = 1 - \frac{sk}{2!} + \frac{(sk)^2}{3!} + \cdots$$

573

因此，当 $k \to 0$ 时，该拉普拉斯变换的值趋近于 1。即

$$\mathcal{L}\{\delta(t)\} = 1$$

因为上面脉冲的面积是 1，所以我们可以定义该脉冲的大小为 1。以上给出了单位脉冲函数的拉普拉斯变换。同理可以推出大小为 a 的脉冲 $a\delta(t)$ 的拉普拉斯变换为：

$$\mathcal{L}\{a\delta(t)\} = a$$

A.3　标准拉普拉斯变换

函数的拉普拉斯变换通常并不需要计算积分，因为有现成的拉普拉斯变换表格给出了常见函数的拉普拉斯变换。我们结合一些拉普拉斯变换的性质知识（见下一节），就基本上可以解决通常遇到的问题。表 A-1 列出了一些常见的时间函数及其拉普拉斯变换。注意：假设时间函数 $f(t) = 0(t < 0)$ 而且该表从大多数时间函数中略去了与 $u(t)$ 相关的项。

表 A-1　拉普拉斯变换

时间函数 $f(t)$	拉普拉斯变换 $F(s)$
$\delta(t)$，单位脉冲	1
$\delta(t-T)$，延时单位脉冲	e^{-sT}
$u(t)$，单位阶跃	$\frac{1}{s}$
$u(t-T)$，延时单位阶跃	$\frac{\mathrm{e}^{-sT}}{s}$
t，单位斜坡	$\frac{1}{s^2}$
t^n，n 次方斜坡	$\frac{n!}{s^{n+1}}$
e^{-at}，指数衰减	$\frac{1}{s+a}$
$1-\mathrm{e}^{-at}$，指数增长	$\frac{a}{s(s+a)}$
$t\mathrm{e}^{-at}$	$\frac{1}{(s+a)^2}$
$t^n\mathrm{e}^{-at}$	$\frac{n!}{(s+a)^{n+1}}$
$t - \frac{1-\mathrm{e}^{-at}}{a}$	$\frac{a}{s^2(s+a)}$
$\mathrm{e}^{-at} - \mathrm{e}^{-bt}$	$\frac{b-a}{(s+a)(s+b)}$

574

（续）

时间函数$f(t)$	拉普拉斯变换$F(s)$
$(1-at)e^{-at}$	$\frac{s}{(s+a)^2}$
$1-\frac{b}{b-a}e^{-at}+\frac{a}{b-a}e^{-bt}$	$\frac{ab}{s(s+a)(s+b)}$
$\frac{e^{-at}}{(b-a)(c-a)}+\frac{e^{-bt}}{(c-a)(a-b)}+\frac{e^{-ct}}{(a-c)(b-c)}$	$\frac{1}{(s+a)(s+b)(s+c)}$
$\sin\omega t$，正弦波	$\frac{\omega}{s^2+\omega^2}$
$\cos\omega t$，余弦波	$\frac{s}{s^2+\omega^2}$
$e^{-at}\sin\omega t$，减幅正弦波	$\frac{\omega}{(s+a)^2+\omega^2}$
$e^{-at}\cos\omega t$，减幅余弦波	$\frac{s+a}{(s+a)^2+\omega^2}$
$1-\cos\omega t$	$\frac{\omega^2}{s(s^2+\omega^2)}$
$t\cos\omega t$	$\frac{s^2-\omega^2}{(s^2+\omega^2)^2}$
$t\sin\omega t$	$\frac{2\omega s}{(s^2+\omega^2)^2}$
$\sin(\omega t+\theta)$	$\frac{\omega\cos\theta+s\sin\theta}{s^2+\omega^2}$
$\cos(\omega t+\theta)$	$\frac{s\cos\theta-\omega\sin\theta}{s^2+\omega^2}$
$\frac{\omega}{\sqrt{1-\zeta^2}}e^{-\zeta\omega t}\sin\omega\sqrt{1-\zeta^2}t$	$\frac{\omega^2}{s^2+2\zeta\omega s+\omega^2}$
$1-\frac{1}{\sqrt{1-\zeta^2}}e^{-\zeta\omega t}\sin(\omega\sqrt{1-\zeta^2}t+\varphi)$，$\cos\varphi=\zeta$	$\frac{\omega^2}{s(s^2+2\zeta\omega s+\omega^2)}$

A.3.1 拉普拉斯变换的性质

本节概括介绍拉普拉斯变换的基本性质。由于具备这些性质，所以使表 A-1 中的标准拉普拉斯变换可以广泛用于各种情况。

575

1. 线性性质

对于两个独立的时间函数，例如$f(t)$ 和$g(t)$，其和的拉普拉斯变换等于二者独立的拉普拉斯变换的和：

$$\mathcal{L}\{af(t)+bg(t)\}=a\mathcal{L}f(t)+b\mathcal{L}g(t)$$

a 和 b 均为常量。因此，对于如 $1+2t+4t^2$ 这样的函数求拉普拉斯变换，可以由各项拉普拉斯变换的和给出。在表 A-1 中查第 3、5、6 条公式可知，结果为

$$F(s)=\frac{1}{s}+\frac{2}{s^2}+\frac{8}{s^3}$$

2. s 域位移性质

该性质有时也称**第一位移性质**，用于对具有指数因子的函数进行拉普拉斯变换。若 $F(s)=\mathcal{L}\{f(t)\}$，则有

$$\mathcal{L}\{e^{at}f(t)\}=F(s-a)$$

例如，对函数 $e^{at}t^n$，由于函数 t^n 的拉普拉斯变换可由表 A-1 中的第 6 条公式给出，即 $n!/s^{n+1}$，则整个函数的拉普拉斯变换为

$$\mathcal{L}\{e^{at}t^n\}=\frac{n!}{(s-a)^{n+1}}$$

3. 时域位移性质

如果一个信号的延迟时间为 T，其拉普拉斯变换需要乘以因子 e^{-sT}。如果 $F(s)$ 是 $f(t)$ 的拉普拉斯变换，则

$$\mathcal{L}\{f(t-T)u(t-T)\} = e^{-sT}F(s)$$

这种情况称为**第二位移定理**。

时域位移特性可以应用于所有的拉普拉斯变换。因此，对于一个脉冲$\delta(t)$延迟了时间 T 给出函数 $\delta(t-T)$，则脉冲 $\delta(t)$ 的拉普拉斯变换，即 1，乘以 e^{-sT}得到 $1e^{-sT}$，即是延迟函数的拉普拉斯变换。

4. 周期函数

对于一个周期为 T 的周期函数 $f(t)$，其拉普拉斯变换为

$$\mathcal{L}f(t) = \frac{1}{1-e^{-sT}}F_1(s)$$

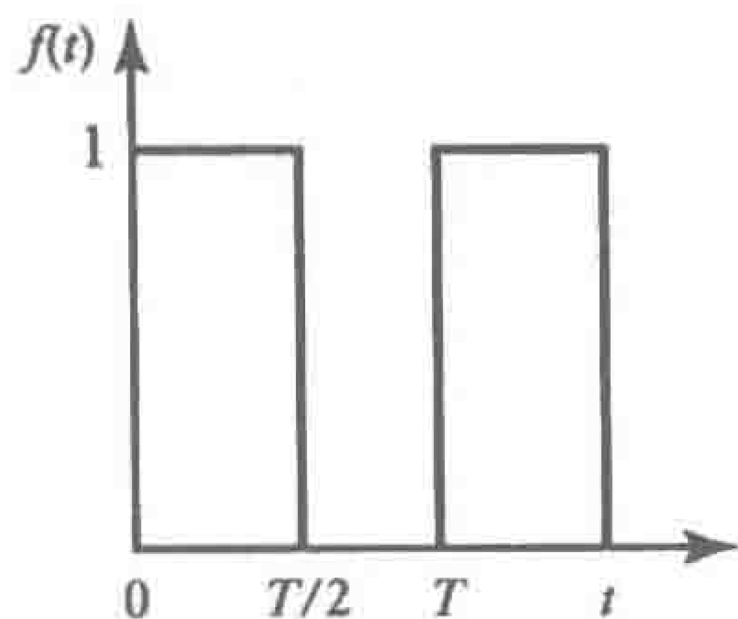


图 A-3 矩形脉冲

$F_1(s)$ 是该函数在第一个周期的拉普拉斯变换。因此，要求一个周期为 T 的矩形脉冲序列的拉普拉斯变换，如图 A-3 所示，可以先求出单个矩形脉冲的拉普拉斯变换，即 $(1/s)(1-e^{-sT/2})$。根据前面的公式，拉普拉斯变换为：

576

$$\frac{1}{1-e^{-sT}} \times \frac{1}{s}(1-e^{-sT/2}) = \frac{1}{s(1+e^{-sT/2})}$$

5. 初值与终值定理

初值定理：若函数 $f(t)$ 的拉普拉斯变换为 $F(s)$，则当时间 t 趋近于 0 时，函数的极限值满足下式：

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s)$$

例如，函数的拉普拉斯变换为 $3/(s+2)$，当 s 趋近于无穷时，$3s/(s+2)=3/(1+2/s)$ 取极限为 3，则该函数的初值为 3。

终值定理：若函数 $f(t)$ 的拉普拉斯变换为 $F(s)$，则当时间 t 趋近于无穷大时，函数值满足下式：

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$$

6. 微分定理

函数 $f(t)$ 微分的拉普拉斯变换为：

$$\mathcal{L}\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0)$$

其中 $f(0)$ 为函数在 $t=0$ 时刻的值。例如，$2(dx/dt)+x=4$ 的拉普拉斯变换为 $2[sX(s)-x(0)]+X(s)=4/s$，如果在 $t=0$ 时 $x=0$，则上式可写为 $2sX(s)+X(s)=4/s$ 或者 $X(s)=4/[s(2s+1)]$。

对于二阶导数，则有

$$\mathcal{L}\left\{\frac{d^2}{dt^2}f(t)\right\} = s^2F(s) - sf(0) - \frac{d}{dt}f(0)$$

其中 $df(0)/dt$ 是 $t=0$ 时函数 $f(t)$ 的一阶导数的值。

7. 积分定理

对于函数 $f(t)$，其积分函数的拉普拉斯变换满足下式：

$$\mathcal{L}\left\{\int_0^t f(t)\,dt\right\} = \frac{1}{s}F(s)$$

例如，函数 e^{-t}在区间 $0\sim t$ 上积分的拉普拉斯变换为：

577

$$L\left\{\int_0^t e^{-t}dt\right\} = \frac{1}{s}\mathcal{L}\{e^{-t}\} = \frac{1}{s(s+1)}$$

A.4 拉普拉斯反变换

拉普拉斯反变换是指将拉普拉斯变换 $F(s)$ 转换为时间函数 $f(t)$。该运算可以写为下式：

$$\mathcal{L}^{-1}\{F(s)\} = f(t)$$

反变换通常可通过表 A-1 查取。利用拉普拉斯变换的线性性质，即一个拉普拉斯变换是两个单独的拉普拉斯变换之和，那么可以先求两个单独的拉普拉斯反变换，再将其求和，即得到要求的拉普拉斯反变换，如下式所示：

$$\mathcal{L}^{-1}\{aF(s) + bG(s)\} = a\mathcal{L}^{-1}F(s) + b\mathcal{L}^{-1}G(s)$$

因此，在求反变换时，通常将拉普拉斯变换函数整理成表 A-1 中的标准形式。例如求 $3/(2s+1)$ 的反变换，将其整理成如下形式：

$$\frac{3(1/2)}{s+(1/2)}$$

表 A-1 中第 7 项拉普拉斯变换函数 $1/(s+a)$ 的反变换函数为 e^{-at}。则所要求的反变换为该式乘以常量 $(3/2)$，$a=(1/2)$，即 $(3/2)\ e^{-t/2}$。

另一个例子，对 $(2s+2)/(s^2+1)$ 求拉普拉斯反变换，可先将其整理为：

$$2\left[\frac{s}{s^2+1} + \frac{1}{s^2+1}\right]$$

方括号内的第一项有反变换 $\cos t$（表 A-1 中第 17 项），第二项有反变换 $\sin t$（表 A-1 中第 16 项）。则总的拉普拉斯反变换为 $2\cos t + 2\sin t$。

A.4.1 部分分式

通常情况下，$F(s)$ 是两个多项式的比，不能简单地在表 A-1 中找到对应的标准变换。因此在使用标准变换前，必须首先将其转换为简单分数的形式。把一个表达式转换为简单分数项的过程称为**部分分式**。如果分子的次数小于分母的次数，则可以使用该方法。多项式的次数是指其表达式中 s 的最高次幂。若分子次数大于或等于分母的次数，则必须将分子除以分母，直到结果为各分项之和的形式，其中余数项分子的次数低于分母的次数。

部分分式可归纳为 3 种类型：

1）分母包含唯一形式的因式，且形式如 $(s+a)$、$(s+b)$、$(s+c)$ 等，表达式的形式为：

578

$$\frac{f(s)}{(s+a)(s+b)(s+c)}$$

其部分分式记为：

$$\frac{A}{(s+a)} + \frac{B}{(s+b)} + \frac{C}{(s+c)}$$

2）分母的因式 $(s+a)$ 重复，即分母包含该因式的多次幂，表达式的形式为：

$$\frac{f(s)}{(s+a)^n}$$

其部分分式记为：

$$\frac{A}{(s+a)^1} + \frac{B}{(s+a)^2} + \frac{C}{(s+a)^3} + \cdots + \frac{N}{(s+a)^n}$$

3）分母中含有 2 次因式，且该因式无法在实数域内分解，表达式的形式为：

$$\frac{f(s)}{(as^2+bs+c)(s+d)}$$

其部分分式记为：

$$\frac{As+B}{as^2+bs+c}+\frac{C}{s+d}$$

常量 A、B、C 等可以利用原表达式和部分分式恒等（不论 s 取何值）或者 s^n 的系数相等的方法求得。第一种求取方法可用下面的例子说明：

$$\frac{3s+4}{(s+1)(s+2)}$$

其部分分式为

$$\frac{A}{s+1}+\frac{B}{s+2}$$

因为两边相等，所以有下列等式：

$$\frac{3s+4}{(s+1)(s+2)}=\frac{A(s+2)+B(s+1)}{(s+1)(s+2)}$$

因此，$3s+4=A(s+2)+B(s+1)$，不论 s 取何值。下面取 s 的值使含某未知常量的项为 0，从而确定另一常量的值。因此不妨取 $s=-2$，得到 $3(-2)+4=A(-2+2)+B(-2+1)$，推出 $B=2$。如果令 $s=-1$，则有 $3(-1)+4=A(-1+2)+B(-1+1)$，推出 $A=1$。因此

$$\frac{3s+4}{(s+1)(s+2)}=\frac{1}{s+1}+\frac{2}{s+2}$$

579

习题

A. 1　求出下列式子的拉普拉斯变换。

(a) $2t$。

(b) $\sin 2t$。

(c) 单位脉冲函数（当 $t=2$ 时）。

(d) $4\mathrm{d}x/\mathrm{d}t$，当 $t=0$ 时，$x=2$。

(e) $3\mathrm{d}^2x/\mathrm{d}t^2$，当 $t=0$ 时 $x=0$，$\mathrm{d}x/\mathrm{d}t=0$。

(f) e^{-t}在$0\sim t$之间积分。

A. 2　求下式的拉普拉斯反变换。

(a) $1/s^2$

(b) $5s/(s^2+9)$

(c) $(3s-1)/[s(s-1)]$

(d) $1/(s+3)$

A. 3　求函数的初值，该函数的拉普拉斯变换为 $5/(s+2)$。

580

附录B 数字系统

B.1 数字系统

十进制系统是以0、1、2、3、4、5、6、7、8、9十个符号或数字为基础的。当用十进制系统表示一个数的时候，每个数字在这个数中所在的位置表示了它的权，权从右到左依次乘以一个因子10，如下所示。

…	10^3	10^2	10^1	10^0
	千	百	十	个

二进制系统是以0和1两个符号或数字为基础的，它们叫作二进制数或二进制位，当一个数用二进制系统表示的时候，每个数字在这个数中所在的位置表示了它的权，权从右到左依次乘以一个因子2，如下所示。

…	2^3	2^2	2^1	2^0
	第3位	第2位	第1位	第0位

例如，十进制数15用二进制表示就是1111。在二进制中，第0位叫作**最低有效位**（LSB），最高位叫作**最高有效位**（MSB）。

八进制系统是以0，1，2，3，4，5，6，7八个数字为基础的。当一个数用八进制系统表示的时候，每个数字在这个数中所在的位置表示了它的权，权从右到左依次乘以一个因子8，如下所示。

$$\cdots \quad 8^3 \quad 8^2 \quad 8^1 \quad 8^0$$

例如，十进制数15用八进制表示为17。

十六进制系统是以0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F十六个符号或数字为基础的，当一个数用十六进制系统表示的时候，每个数字在这个数中所在的位置表示了它的权，权从右到左依次乘以一个因子16，如下所示。

$$\cdots \quad 16^3 \quad 16^2 \quad 16^1 \quad 16^0$$

例如，十进制数15用十六进制表示为F。十六进制在微处理器编程中广泛应用，因为在输入数据时，这种表示方法非常紧凑。

二进制编码的十进制系统（BCD系统）在计算机系统中应用广泛。每位十进制数字分别用二进制进行编码。例如十进制数15用BCD表示为00010101。对于输出必须是十进制显示的微处理器来说，这种编码对于其输出是非常有用的，要显示的每个十进制位由微处理器自身的二进制编码提供。

581

表B-1 数字系统

十进制	二进制	BCD码	八进制	十六进制
0	0000	0000 0000	0	0
1	0001	0000 0001	1	1
2	0010	0000 0010	2	2
3	0011	0000 0011	3	3

（续）

十进制	二进制	BCD 码	八进制	十六进制
4	0100	0000 0100	4	4
5	0101	0000 0101	5	5
6	0110	0000 0110	6	6
7	0111	0000 0111	7	7
8	1000	0000 1000	10	8
9	1001	0000 1001	11	9
10	1010	0001 0000	12	A
11	1011	0001 0001	13	B
12	1100	0001 0010	14	C
13	1101	0001 0011	15	D
14	1110	0001 0100	16	E
15	1111	0001 0101	17	F

表 B-1 给出了数字在十进制、二进制、BCD 码、八进制和十六进制的例子。

B.2 二进制运算

二进制加法遵循下面的规则：

$$0+0=0$$

$$0+1=1+0=1$$

$1+1=10$，即 0 + 进位 1

$1+1+1=11$，即 1 + 进位 1

十进制数 $14+19=33$，在二进制中变成了：

被加数	01110
加数	10011
和	100001

第 0 位为 $0+1=1$，第 1 位为 $1+1=10$，向下一位进 1，此位为 0。第 3 位为 $1+0$ + 进位 $1=10$，第 4 位为 $0+1$ + 进位 $1=10$。继续进行所有的位，最后的和加上进位 1，最终结果为 100001。二进制数 A 加上 B 得到 C，即 $A+B=C$，那么 A 是**被加数**、B 是**加数**、C 是和。

二进制减法遵循下面的规则：

$$0-0=0$$

$$1-0=1$$

$$1-1=0$$

$0-1=10-1$ + 借 $1=1$ + 借 1

582

当计算 $0-1$ 时，需要向左边包含 1 的下一列借 1，如下面的例子，$27-14=13$：

被减数	11011
减数	01110
差	01101

第 0 位为 $1-0=1$，第 1 位为 $1-1=0$，第 2 位为 $0-1$，需要向下一位借 1，得到 $10-1=1$，第 3 位为 $0-1$，因为已借出 1，$10-1=1$，第 4 位 $0-0=0$，也因为已借出 1。二进制数 A 减去 B 得到 C，即 $A-B=C$，那么 A 是**被减数**，B 是**减数**，C 是**差**。

当使用另外一种减法算法时，在电子上二进制数减法会更加容易实现。上面的例子可以理解为一个正数与一个负数相加。下面我们就介绍怎么样规定负数以使减法转化为加法。这种方法使我们能处理任何情况下的负数。

目前所处理的数据都是**无符号数**，因为数据本身没有包含正、负信息。对于**有符号数**，最高有效位是它的标志位，0代表正数，1代表负数。如果我们要用正数，就先将它写成一般形式，然后在最高位前补0，同样使用负数就补1。所以二进制的正数10010要写成010010，负数形式就是110010。但是对于计算机处理数据方便性来说，这并不是表示负数的最好方法。

表示负数的一个更有效的方式是用二进制的补码方法。一个二进制数有两个补数，称为**反码和补码**。二进制反码就是将二进制数的非符号位的所有0全改为1，所有1全改为0。二进制补码由二进制反码加1得到。对于负数，先得到二进制补码，然后将其符号位置1，对于正数，则将其符号位置0。以十进制数 -3 为例，将它写成有符号的二进制补码形式。首先将无符号数3写成二进制形式0011，然后得到其反码1100，将反码加1得到无符号补码1101，最后将符号位置1表明它是负数，因此结果为11101。下面是另外一个例子，得到 -6 的有符号的8位二进制补码。

无符号二进制数	000 0110
反码	111 1001
加1	1
无符号二进制补码	111 1010
有符号二进制补码	1111 1010

583

表 B-2　有符号数

十进制数	有符号数		十进制数	有符号数	
+127	0111 1111	仅仅是符号位为0的二进制数	-1	1111 1111	符号位为1的二进制补码
…			-2	1111 1110	
+6	0000 0110		-3	1111 1101	
+5	0000 0101		-4	1111 1100	
+4	0000 0100		-5	1111 1011	
+3	0000 0011		-6	1111 1010	
+2	0000 0010		…		
+1	0000 0001		-127	1000 0000	
+0	0000 0000				

对于一个正数，我们将它写成普通二进制形式，并在最高位前加0。因此，正二进制数100 1001将会写成0100 1001。表B-2给出了二进制系统中一些数的例子。

两个正数相减可以看成是，首先得到减数的有符号补码，然后将其加到有符号的被减数上。因此，十进制数4减去十进制6得到：

有符号被减数	0000 0100
减数，有符号补码	1111 1010
和	1111 1110

上例中结果最高有效位为1，所以结果是一个负数，这是 -2 的有符号的补码。

考虑另一个例子，57 -43。正数57的有符号数为0011 1001，-43的有符号补码通过下式得到：

43的无符号二进制数	010 1011
反码	101 0100
加1	1
无符号补码	101 0101
有符号补码	1101 0101

因此，通过有符号的正数和有符号的补码相加得到结果

有符号的被减数	0011 1001
减数，有符号的补码	1101 0101
和	0000 1110 + 进位 1

进位1被忽略，因此结果为0000 1110。由于最高有效位是0，所以结果是正数，为十进制数14。

如果要进行两个负数相加，那么需要得到它们的有符号补码，并相加。当一个数为负数时，我们需要用到它的有符号补码；如果一个数为正数，我们只需要用它的有符号二进制数。 584

B.3 浮点数

在十进制系统中，较大的数，比如120 000，常常写成科学记数形式1.2×10^5或120×10^3；较小的数，比如0.000 120，则记为1.2×10^{-4}，而不是将小数点的位置固定。用这种方式记数，数经常写成10的幂形式。同样，我们也可以对二进制数用这种记数方式，但它们要写成2的幂形式。例如，我们将1010写成1.010×2^3或者10.10×2^2。因为可以通过选择2的幂次数来移动二进制小数点的位置，这种记数法就叫作**浮点记数法**。

浮点数的形式为$a\times r^e$，其中a是**尾数**、r是**基数**、e是**指数或幂**。在二进制中，基数就是2，也就是$a\times2^e$。与定点数表示相比，浮点数的好处就是用相同位数的数字可以表示更大范围的数。

由于浮点数可能用不同的形式来存储一个数，例如0.1×10^2和0.01×10^3，所以用计算系统将这些数字规范化了，即它们都写为$0.1\times r^e$的形式。因此，对于二进制数，则写成0.1×2^e的形式，0.00001001记为0.1001×2^{-4}。考虑到二进制数的符号，需要对正数添加符号位0，对负数添加符号位1。所以数0.1001×2^{-4}若为负数则记为1.1001×2^{-4}，若为正数则记为0.1001×2^{-4}。

如果要进行2.01×10^3和10.2×10^2相加运算，必须使它们的幂（通常用术语指数）相同，所以写成$2.01\times10^3+1.02\times10^3$。对它们进行按位相加，并将进位考虑进去得到$3.03\times10^3$。对于二进制浮点数采用相同的步骤，所以要进行$0.101100\times2^4$和$0.111100\times2^2$的相加运算，首先要统一它们的指数，如$0.101100\times2^4$和$0.001111\times2^4$，然后对它们进行按位相加得到$0.111011\times2^4$。

减法也类似，只有当两个浮点数的指数相等的时候才能对它们进行按位相减。因此0.1101100×2^{-4}减去0.1010100×2^{-5}可以写成$0.1101100\times2^{-4}-0.0101010\times2^{-4}$，结果是$0.1000010\times2^{-4}$。

B.4 格雷码

我们来看两个连续的二进制数0001和0010（十进制的1和2），在从0001到0010的变换中，有两位发生了改变，因此，对于绝对编码器（见2.3.7节）来说，连续的位置对应着连续的二进制码，这种情况下二进制码会有两位发生变化。这两位的变化必须同时进行，否则会出现问题。如果一位稍微超前于另一位变化，那么这个二进制码会短时变为另外一个码。所以从0001到0010变化时，会短时出现0011或0000，因此可以用另外一种编码方式。

格雷码（Gray code）就是这样一种编码：由一个二进制码到相邻码变化时，在编码组中只有一位发生变化。格雷码没有位权，因为在编码组中没有给位分配具体的权重。因此把它作为算法进行操作是不合适的，但是它广泛应用于像绝对编码器这样的输入/输出设备中。表B-3 585

列出了十进制数和其在二进制和格雷码中的数值。

表 B-3　格雷码

十进制数	二进制码	格雷码	十进制数	二进制码	格雷码
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

习题

B.1　8 位二进制数能表示的最大十进制数是多少?

B.2　将下列二进制数转换为十进制数：(a) 1011；(b) 10 0001 0001。

B.3　将下列十进制数转换为十六进制数：(a) 423；(b) 529。

B.4　将下列 BCD 码转换为十进制数：(a) 0111 1000 0001；(b) 0001 0101 0111。

B.5　写出下列十进制数的补码：(a) −90；(b) −35。

B.6　下列数的偶校验位是什么：(a) 100 1000；(b) 100 1111。

586

B.7　用补码来计算下面的十进制减法：(a) 21 − 13；(b) 15 − 3。

附录C 布尔代数

C.1 布尔代数的定律

布尔代数中用到了二进制数位1和0，以及运算·、+和“非”。代数定律为：

1）任何二进制数和其自身的逻辑或（OR）是其自身：$A+A=A$。

2）任何二进制数和其自身的逻辑与（AND）是其自身：$A\cdot A=A$。

3）逻辑或和逻辑与满足交换率，即

$$A+B=B+A,\quad A\cdot B=B\cdot A$$

4）由下列真值表可知：

$$A+(B\cdot C)=(A+B)\cdot(A+C)$$

A	**B**	**C**	**B·C**	**A+B·C**	**A+B**	**A+C**	**(A+B)·(A+C)**
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

5）同样，利用真值表可知它和普通代数一样满足分配率：

$$A\cdot(B+C)=A\cdot B+A\cdot C$$

6）任何二进制数与其逻辑非的逻辑或为1：

$$A+\overline{A}=1$$

7）任何二进制数与其逻辑非的逻辑与为0：

$$A\cdot\overline{A}=0$$

8）任何二进制数与0的逻辑或为其自身，与1的逻辑或为1，即

$$A+0=A,\quad A+1=1$$

9）任何二进制数与0的逻辑与为0，与1的逻辑与为其自身，即

$$A\cdot 0=0,\quad A\cdot 1=A$$

作为说明，下面介绍用上述原则来化简：

$$(A+B)\cdot\overline{C}+A\cdot C$$

首先用第5条原则，有

$$A\cdot\overline{C}+B\cdot\overline{C}+A\cdot C$$

整理上式，然后用第6条原则，有

$$A\cdot(\overline{C}+C)+B\cdot\overline{C}=A\cdot 1+B\cdot\overline{C}$$

因此，使用第9条原则后，表达式简化为：

$$A+B\cdot\overline{C}$$

C.2 德·摩根定律

如上所示，布尔代数原则可以简化布尔表达式，同时还有一项原则——**德·摩根定律**（De Morgan's law）：

1）A 和 B 的逻辑或再逻辑非与其各自逻辑非再逻辑与相等，下列真值表证实了这一结论：

$$\overline{A+B}=\overline{A}\cdot\overline{B}$$

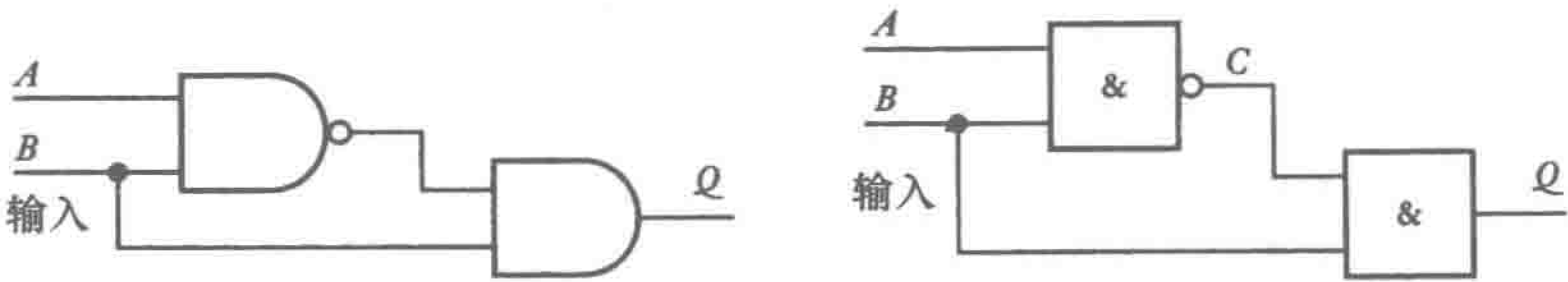
A	B	$A+B$	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}\cdot\overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

2）A 和 B 的逻辑与的逻辑非与其各自逻辑非再逻辑或相等，下列真值表证实了这一结论：

$$\overline{A\cdot B}=\overline{A}+\overline{B}$$

A	B	$A\cdot B$	$\overline{A\cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

为了说明德·摩根定理，考虑图 C-1 中的逻辑电路图的简化。



588

图 C-1 电路简化 1

输出对输入的布尔方程为：

$$Q=\overline{A\cdot B}\cdot B$$

应用上面的第二个定理，变换为：

$$Q=(\overline{A}+\overline{B})\cdot B$$

从而得到

$$Q=\overline{A}\cdot B+\overline{B}\cdot B=\overline{A}\cdot B+0=\overline{A}\cdot B$$

因此简化的逻辑电路如图 C-2 所示。

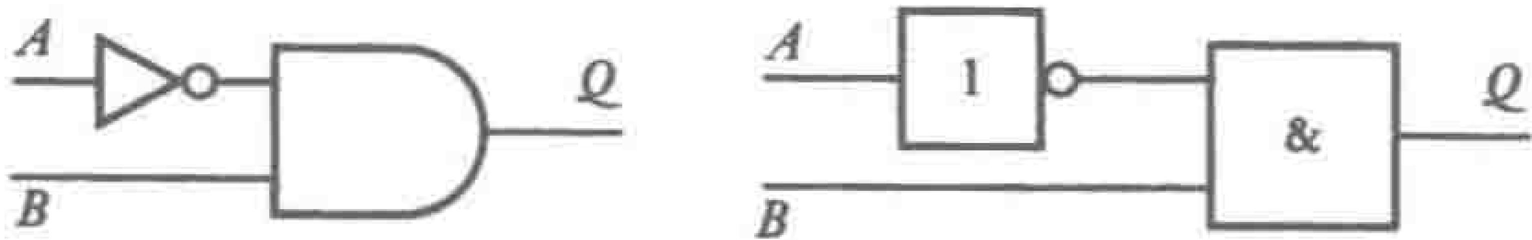


图 C-2 电路简化 2

C.3 由真值表产生布尔函数

如果系统的要求可用真值表来规定，那怎样才能设计出最少门的逻辑系统来给出这样的真值表呢？

布尔代数一般用于将开关功能转换成一些等价的形式，有些需要用更多的逻辑门。然而，可用与门来驱动单个或门（或相反）来简化这些形式。两个与门驱动一个或门（见

图 C-3a）为：

$$A \cdot B + A \cdot C$$

这称为**积和**（sum of product）形式。

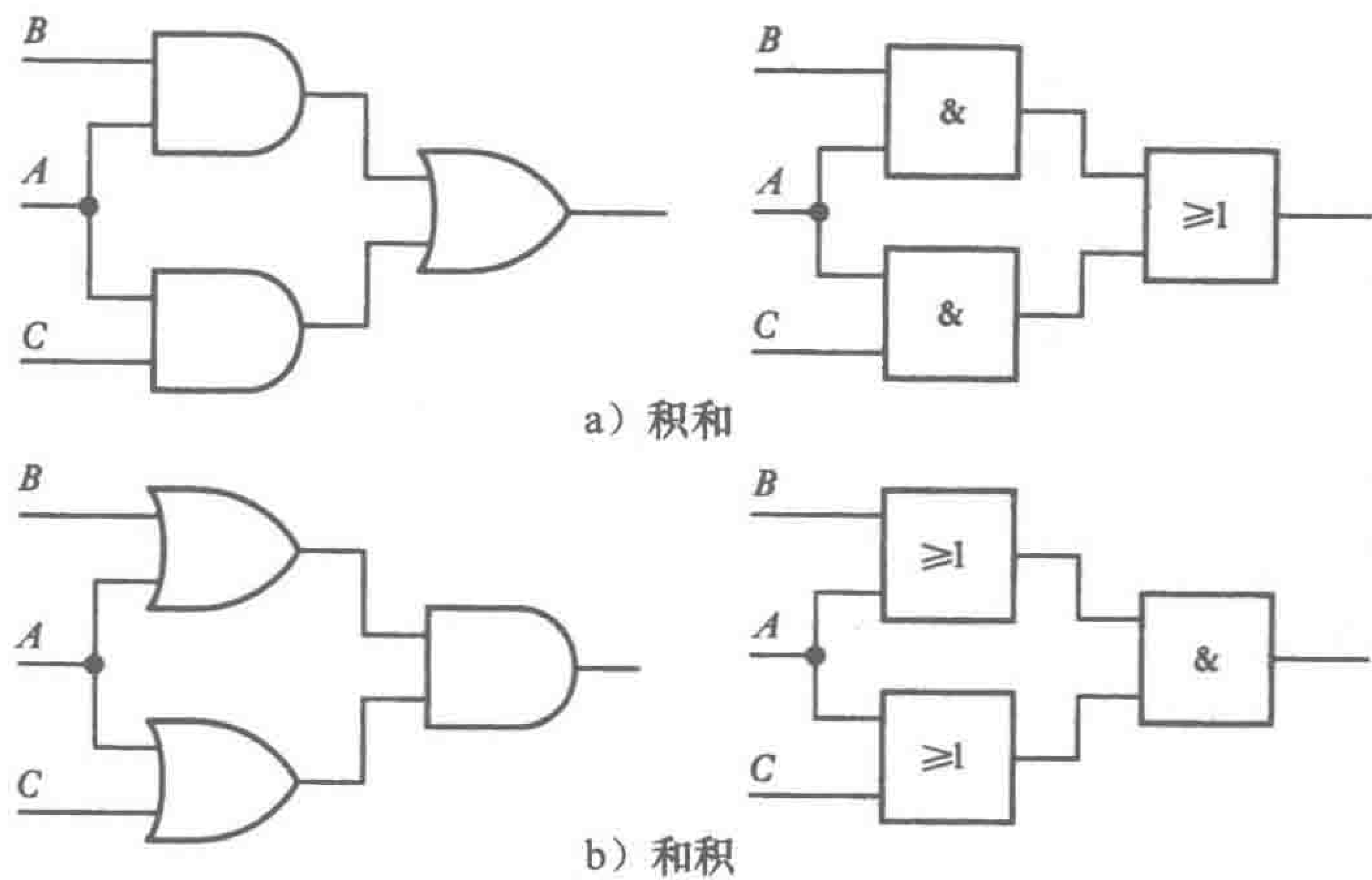


图 C-3 “积和”与“和积”

图 C-3b 中是两个或门驱动一个与门，即

$$(A + B) \cdot (A + C)$$

这称为**和积**（product of sum）形式。因此考虑哪个最小形式对应所给的真值表时，通常的方法是找到“积和”或者“和积”形式来对应数据。通常，积和形式比较常用。所用的步骤要考虑真值表中的每一行，并且找到适应于这一行的积，最后的结果就是将这些积相加的和。 589

假设真值表中有一行：

$$A = 1, B = 0, \quad 且输出 Q = 1$$

当 A 是 1，B 不是 1，则输出为 1，因此满足这一条件的积就是：

$$Q = A \cdot \overline{B}$$

如下表所示，我们可以对真值表的每一行重复这样的操作：

A	B	输出	积
0	0	0	$\overline{A} \cdot \overline{B}$
0	1	0	$\overline{A} \cdot B$
1	0	1	$A \cdot \overline{B}$
1	1	0	$A \cdot B$

然而，只需要考虑当真值的输出是 1 的情况，因为输出 0 对最后的表达式没有影响，所以结果为：

$$Q = A \cdot \overline{B}$$

因此对应于这一真值表的逻辑门系统如图 C-4 所示。

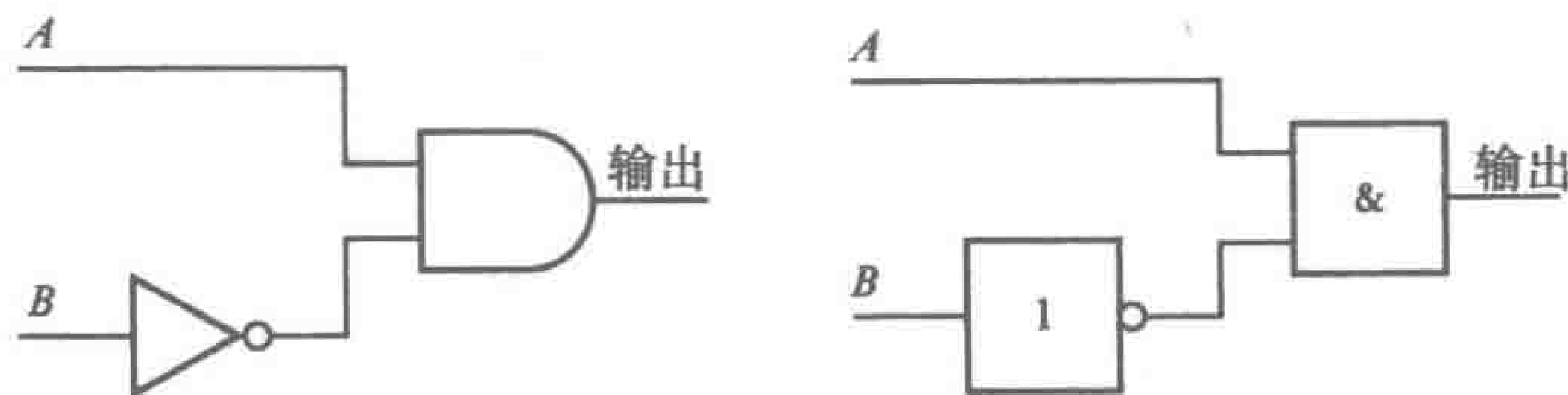


图 C-4 逻辑门真值表

看一个更深入的例子，考虑下面的真值表，它只包括了积输出为1的情况：

A	B	C	输出	积
0	0	0	1	$\overline{A}\cdot\overline{B}\cdot\overline{C}$
0	0	1	0	
0	1	0	1	$\overline{A}\cdot B\cdot\overline{C}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	

590

因此用积和形式表达这个表为：

$$Q=\overline{A}\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot B\cdot\overline{C}$$

可以简化为：

$$Q=\overline{A}\cdot\overline{C}\cdot(\overline{B}+B)=\overline{A}\cdot\overline{C}$$

由此可见，这个真值表可以只用一个NAND门来生成。

C.4 卡诺图

卡诺图（Karnaugh map）是一种图形处理方法，用于对真值表生成的积和形式的布尔表达式进行简化。真值表的每一行有与输入值的组合相对应的输出值。如果只有两个变量，则真值表有4行；3个变量为8行；4个变量为16行。因此两个变量会有4个积项，3个变量有8个，4个变量有16个。卡诺图是一种由方格组成的矩形阵列，每一个格子对应于一个特定的积值。因此，两个输入变量有4个格子，3个输入变量有8个格子，4个输入变量有16个格子。每一行的输出值就放进卡诺图上面对应的格子内，一般只有结果为1时输出值，0时为空。

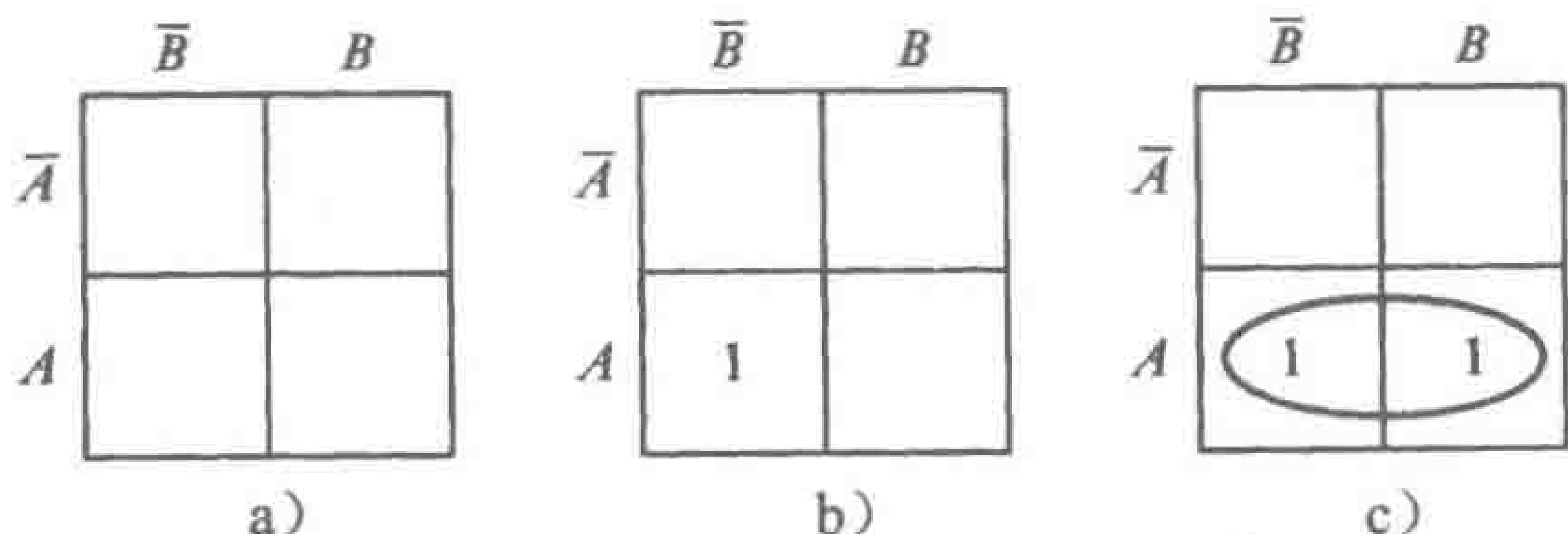
图C-5a为两个变量的卡诺图。每一个格子中的输出由下面的积产生：

左上角为：$\overline{A}\cdot\overline{B}$

左下角为：$A\cdot\overline{B}$

右上角为：$\overline{A}\cdot B$

右下角为：$A\cdot B$



图C-5 两个输入变量的卡诺图

图中方框是这样安排的，平行相连的方框只有一个变量不同，同样，垂直相连的方框也是一样。因此，在两个变量的卡诺图中，水平方向上只有变量A不同，垂直方向上只有变量B不同。

对于下面给出的真值表：

A	B	输出	积
0	0	0	$\overline{A}\cdot\overline{B}$
0	1	0	$\overline{A}\cdot B$
1	0	1	$A\cdot\overline{B}$
1	1	0	$A\cdot B$

如果将计算好的积填入卡诺图中，只有在值为1时填入，值为0时不填，则可得到图C-5b。因为只有左下角的方框里面有真值1，所以真值表可以用布尔表达式表示为：

$$输出 = A\cdot\overline{B}$$

再看一个例子，考虑下面的真值表：

A	B	输出	积
0	0	0	$\overline{A}\cdot\overline{B}$
0	1	0	$\overline{A}\cdot B$
1	0	1	$A\cdot\overline{B}$
1	1	1	$A\cdot B$

它给出了图C-5c中的卡诺图，输出为：

$$输出 = A\cdot\overline{B}+A\cdot B$$

我们可将其简化为：

$$A\cdot\overline{B}+A\cdot B = A\cdot(\overline{B}+B) = A$$

如果包含1的两个元素在垂直方向上有公共边，那么可以将布尔表达式简化为这个公共变量。通过检查卡诺图来实现。由在上面画圈来标注（圈组），如图C-5c所示。

图C-6a是下面真值表的卡诺图，它包含了3个变量：

A	B	C	输出	积
0	0	0	1	$\overline{A}\cdot\overline{B}\cdot\overline{C}$
0	0	1	0	$\overline{A}\cdot\overline{B}\cdot C$
0	1	0	1	$\overline{A}\cdot B\cdot\overline{C}$
0	1	1	0	$\overline{A}\cdot B\cdot C$
1	0	0	0	$A\cdot\overline{B}\cdot\overline{C}$
1	0	1	0	$A\cdot\overline{B}\cdot C$
1	1	0	0	$A\cdot B\cdot\overline{C}$
1	1	1	0	$A\cdot B\cdot C$

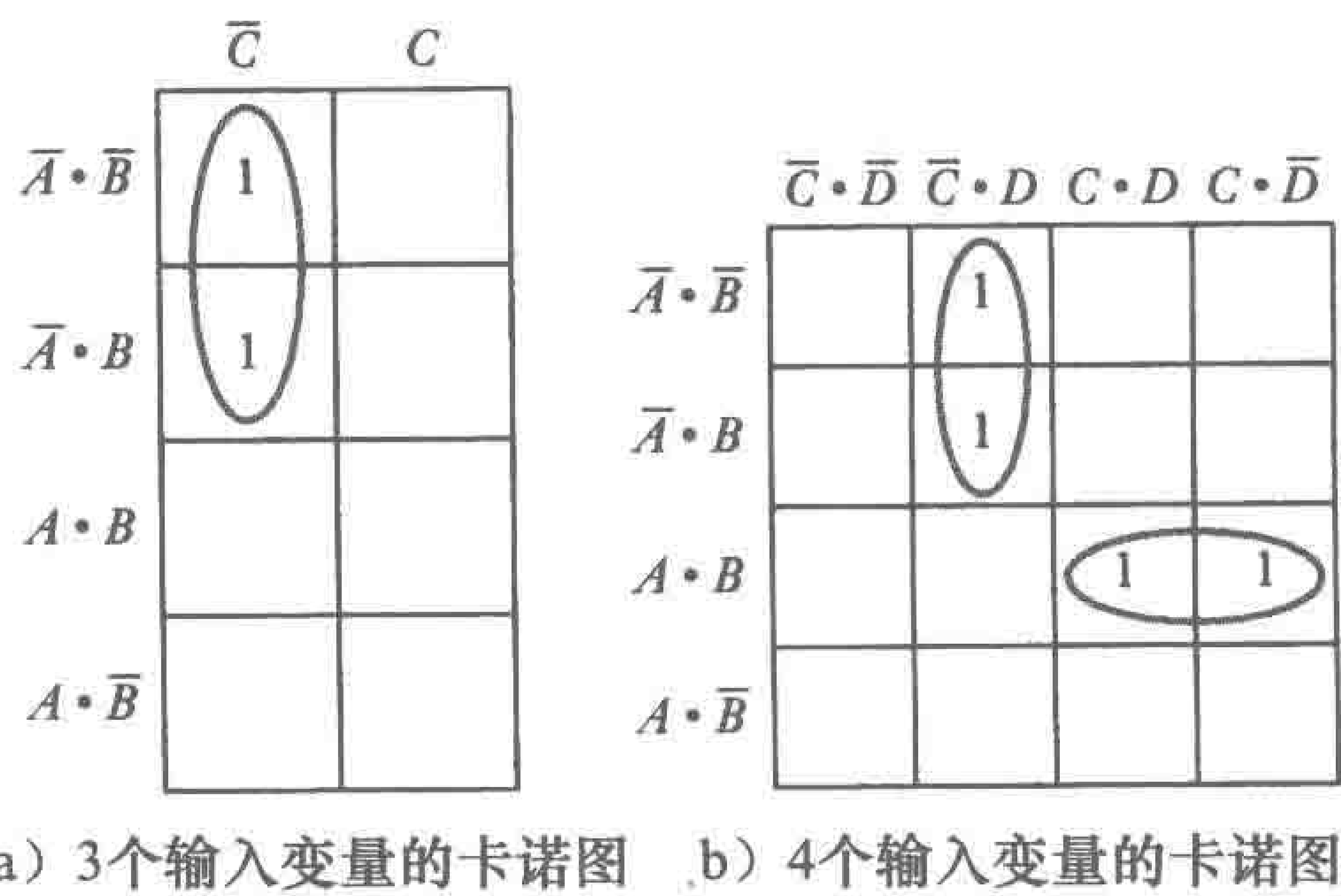
我们可以像前面一样用圈组的方法把结果布尔表达式简化为一般的变量，结果为：

$$输出 = \overline{A}\cdot\overline{C}$$

图C-6b是下面真值表的卡诺图，它含有4个变量，经过画圈标注简化后的结果为：

$$输出 = \overline{A}\cdot\overline{C}\cdot D+A\cdot B\cdot C$$

A	B	C	输出	积
0	0	0	0	
0	0	0	1	$\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D$
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	$\overline{A}\cdot B\cdot\overline{C}\cdot D$
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	$A\cdot B\cdot C\cdot\overline{D}$
1	1	1	1	$A\cdot B\cdot C\cdot D$

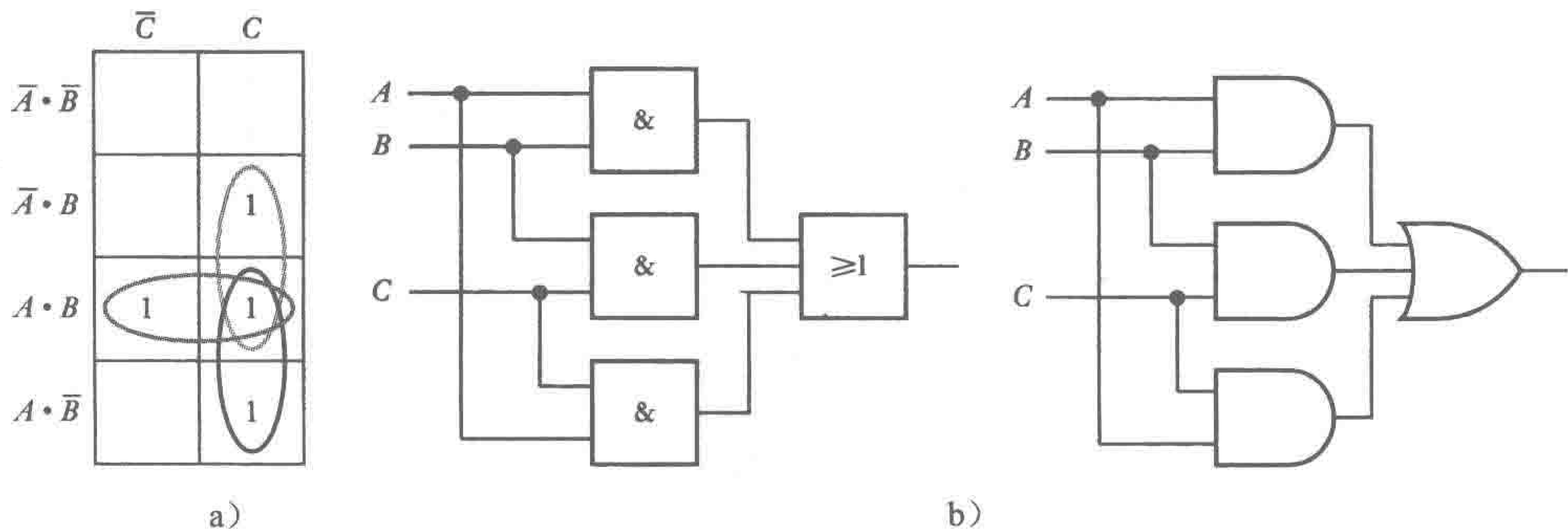


a）3个输入变量的卡诺图　b）4个输入变量的卡诺图

图 C-6　例子

上面讲到的只是关于卡诺图以及圈组使用的一些简单例子。注意，在圈组的过程中，左、右列的上、下行中的元素相邻。图中相对边的方块连在一起。将图中一对相邻格子圈起来，可消去互补形式出现的变量。将 4 个相邻格子圈起来可消去两个互补形式的变量。将 8 个相邻格子圈起来可消去三个互补形式的变量。

为了进一步说明，考虑一台自动机，它只有 A、B、C 三个传感器其中有两个收到信号时才会启动。下面为满足这一要求的真值表，图 C-7a 为这个三变量卡诺图的结果。于是，描述



a）　　　　b）

图 C-7　自动机

机器的传感器输出结果和满足这一图形的布尔表达式为：

$$输出 = A \cdot B + B \cdot C + A \cdot C$$

图 C-7b 是用于产生其布尔表达式的逻辑门电路图。$A \cdot B$ 表示输入 A 和 B 的与门关系，同样，$B \cdot C$ 和 $A \cdot C$ 是另外两个与门。符号“+”表示这三个与门的输出为或门的输入。

591～593

A	B	C	输出	积
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A} \cdot B \cdot C$
1	0	0	0	
1	0	1	1	$A \cdot \overline{B} \cdot C$
1	1	0	1	$A \cdot B \cdot \overline{C}$
1	1	1	1	$A \cdot B \cdot C$

在某些逻辑系统中，有些输入变量组合的输出没有特别指定，称为“无关状态”。当遇到这种情况时，在对应的格子里可以填 1 或者 0，只要方便简化布尔表达式即可。

习题

C.1 给出用来描述下面情形的布尔函数：

(a) 当 A 关闭，并且 B 或 C 有一个关闭时，有输出。

(b) 当 A 或 B 有一个关闭，并且 C 或 D 有一个关闭时，有输出。

(c) 当 A 打开或者 B 关闭时，有输出。

(d) 当 A 打开并且 B 关闭时，有输出。

C.2 说出图 C-8 中每一个逻辑电路的布尔函数。

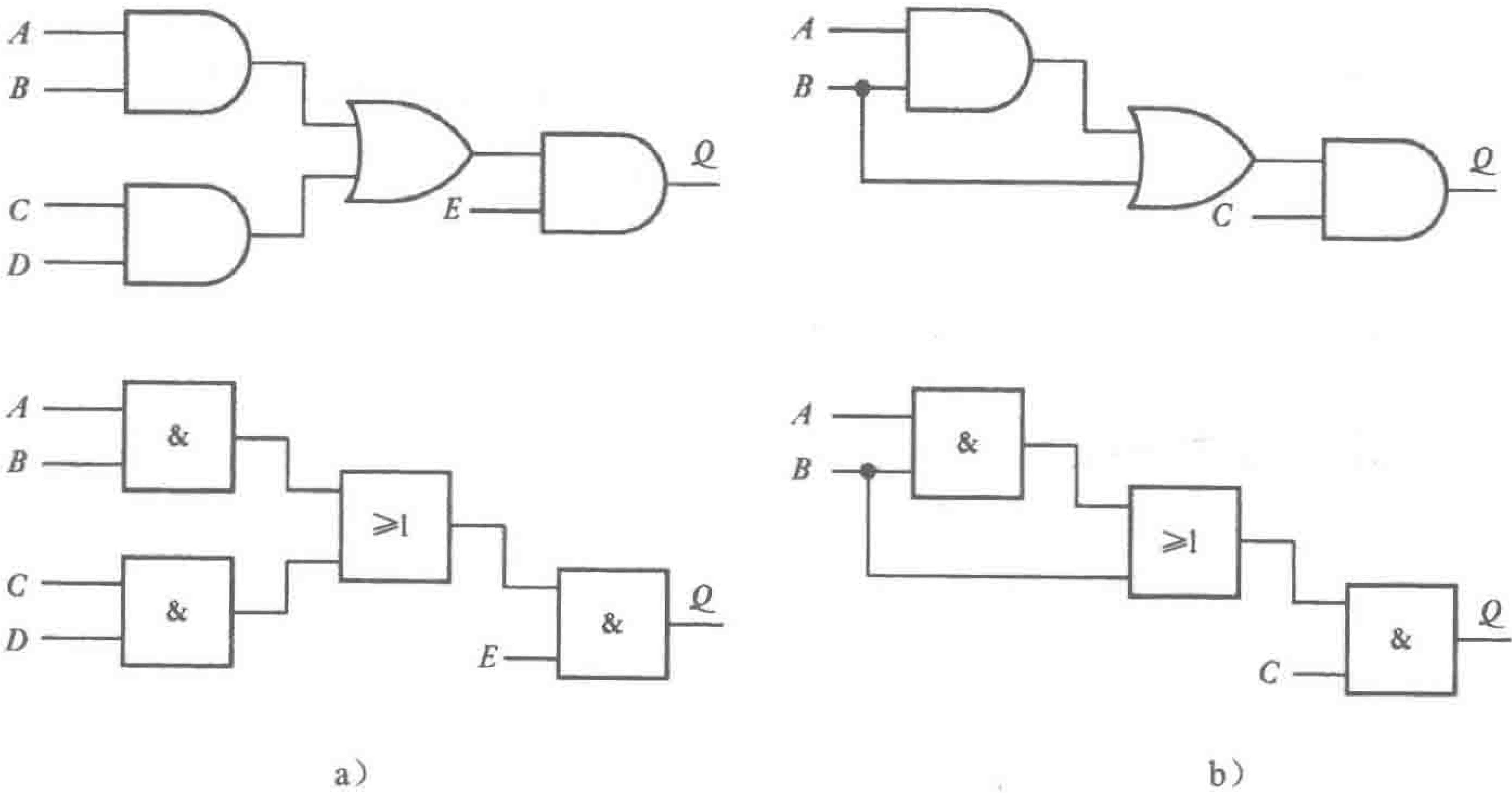


图 C-8 习题 C.2

C.3 建立布尔表达式 $Q = (A \cdot C + B \cdot C) \cdot (A + C)$ 的真值表。

C.4 简化下列布尔表达式：

(a) $Q = A \cdot C + A \cdot C \cdot D + C \cdot D$

(b) $Q = A \cdot \overline{B} \cdot D + A \cdot \overline{B} \cdot \overline{D}$

(c) $Q=A\cdot B\cdot C+C\cdot D+C\cdot D\cdot E$

C.5 用德·摩根定理证明在反向输入的情况下一个 NOR 门等价于一个 AND 门。

C.6 为下面的真值表画出卡诺图，并写出简化后的输出布尔表达式。

(a)

A	*B*	*Q*
0	0	1
0	1	1
1	0	1
1	1	1

(b)

A	*B*	*C*	*Q*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

C.7 用卡诺图简化下面的布尔方程：

(a) $Q=\overline{A}\cdot\overline{B}\cdot C+\overline{A}\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot B\cdot\overline{C}$

(b) $Q=\overline{A}\cdot B\cdot\overline{C}\cdot D+A\cdot\overline{B}\cdot\overline{C}\cdot D+\overline{A}\cdot\overline{B}\cdot\overline{C}\cdot D+A\cdot B\cdot\overline{C}\cdot D+A\cdot B\cdot\overline{C}\cdot\overline{D}+A\cdot B\cdot C\cdot D$

594~595

C.8 设计一个系统，要求有 4 个按钮开关，只有当按对正确的按钮组合时，门才能开，否则警报器响。

附录D 指 令 集

下面是摩托罗拉 M68HC11 单片机、Intel 8051 单片机和 PIC16Cxx 系列微控制器使用的指令集。

M68HC11

指令	助记符	指令	助记符
加载:		将累加器 B 中的位循环右移	RORB
加载累加器 A	LDAA	将存储器中的位向左进行算术移位	ASL
加载累加器 B	LDAB	将累加器 A 中的位向左进行算术移位	ASLA
加载累加器 D	LDD	将累加器 B 中的位向左进行算术移位	ASLB
加载栈指针	LDS	将存储器中的位向右进行算术移位	ASR
加载变址寄存器 X	LDX	将累加器 A 中的位向右进行算术移位	ASRA
加载变址寄存器 Y	LDY	将累加器 B 中的位向右进行算术移位	ASRB
将栈中的数据传送到累加器 A 中	PULA	将存储器中的位向左进行逻辑移位	LSL
将栈中的数据传送到累加器 B 中	PULB	将累加器 A 中的位向左进行逻辑移位	LSLA
将栈中的数据传送到变址寄存器 X 中	PULX	将累加器 B 中的位向左进行逻辑移位	LSLB
将栈中的数据传送到变址寄存器 Y 中	PULY	将累加器 D 中的位向左进行逻辑移位	LSLD
传送寄存器:		将存储器中的位向右进行逻辑移位	LSR
将累加器 A 中数据传送到累加器 B 中	TAB	将累加器 A 中的位向右进行逻辑移位	LSRA
将累加器 B 中数据传送到累加器 A 中	TBA	将累加器 B 中的位向右进行逻辑移位	LSRB
从栈指针传送到变址寄存器 X 中	TSX	将累加器 D 中的位向右进行逻辑移位	LSRD
从栈指针传送到变址寄存器 Y 中	TSY	**根据条件码进行数据测试:**	
从变址寄存器 X 传送到栈指针	TXS	累加器 A 与存储器之间的逻辑测试 AND	BITA
从变址寄存器 Y 传送到栈指针	TYS	累加器 B 与存储器之间的逻辑测试 AND	BITB
交换累加器 D 和变址寄存器 X 中的值	XGDX	比较累加器 A 和累加器 B	CBA
交换累加器 D 和变址寄存器 Y 中的值	XGDY	比较累加器 A 和存储器	CMPA
加/减:		比较累加器 B 和存储器	CMPB
存储器减 1	DEC	比较累加器 D 和存储器	CPD
累加器 A 减 1	DECA	比较变址寄存器 X 和存储器	CPX
累加器 B 减 1	DECB	比较变址寄存器 Y 和存储器	CPY
栈指针减 1	DES	从存储器中减去 $00	TST
变址寄存器 X 减 1	DEX	从累加器 A 中减去 $00	TSTA
变址寄存器 Y 减 1	DEY	从累加器 B 中减去 $00	TSTB
存储器加 1	INC	**中断:**	
累加器 A 加 1	INCA	清除中断屏蔽	CLI
累加器 B 加 1	INCB	置位中断屏蔽	SEI
栈指针加 1	INS	软件中断	SWI
变址寄存器 X 加 1	INX	返回中断	RTI
变址寄存器 Y 加 1	INY	等待中断	WAI
循环移位:		**清除:**	
将存储器中的位循环左移	ROL	清除存储器	CLR
将累加器 A 中的位循环左移	ROLA	清除累加器 A	CLRA
将累加器 B 中的位循环左移	ROLB	清除累加器 B	CLRB
将存储器中的位循环右移	ROR	清除存储器中的位	BCLR
将累加器 A 中的位循环右移	RORA	将存储器中的位置 1	BSET

（续）

指令	助记符	指令	助记符
存储寄存器：		无符号整数除法，D 除以变址寄存器 X	IDIV
保存累加器 A 的内容	STAA	无符号浮点数除法，D 除以变址寄存器 X	FDIV
保存累加器 B 的内容	STAB	**条件转移指令：**	
保存累加器 D 的内容	STD	负则跳转	BMI
保存栈指针寄存器	STS	正则跳转	BPL
保存变址寄存器 X	STX	溢出标志置位，则跳转	BVS
保存变址寄存器 Y	STY	溢出标志清零，则跳转	BVC
将累加器 A 的数据放入栈	PSHA	小于零则跳转	BLT
将累加器 B 的数据放入栈	PSHB	大于等于零跳转	BGE
将变址寄存器 X 的数据放入栈	PSHX	小于等于零跳转	BLE
将变址寄存器 Y 的数据放入栈	PSHY	大于零跳转	BGT
逻辑运算：		相等则跳转	BEQ
和累加器 A 中数据进行逻辑与	ANDA	不相等则跳转	BNE
和累加器 B 中数据进行逻辑与	ANDB	大于则跳转	BHI
和累加器 A 中数据进行异或	EORA	低于或等于则跳转	BLS
和累加器 B 中数据进行异或	EORB	高于或等于则跳转	BHS
和累加器 A 中数据进行逻辑或	ORAA	低于则跳转	BLO
和累加器 B 中数据进行逻辑或	ORAB	进位清零则跳转	BCC
用二进制反码替换存储器中的数据	COM	进位置位则跳转	BCS
用二进制反码替换累加器 A 中的数据	COMA	**转移和跳转：**	
用二进制反码替换累加器 B 中的数据	COMB	跳到地址	JMP
算术运算：		跳到子程序	JSR
将累加器 A 的内容加到 B 中	ABA	从子程序返回	RTS
将累加器 B 的内容加到变址寄存器 X 中	ABX	跳转到子程序	BSR
将累加器 B 的内容加到变址寄存器 Y 中	ABY	总是转移	BRA
将存储器加到累加器 A 中，不带进位	ADDA	从不转移	BRN
将存储器加到累加器 B 中，不带进位	ADDB	若第 n 位 =1 则转移	BRSET
将存储器加到累加器 D 中，不带进位	ADDD	若第 n 位 =0 则转移	BRCLR
将存储器加到累加器 A 中，带进位	ADCA	**状态码：**	
将存储器加到累加器 B 中，带进位	ADCB	清除进位	CLC
十进制调整	DAA	清除溢出位	CLV
从累加器 A 中减去 B 的内容	SBA	进位置位	SEC
从累加器 A 中减去存储器数据，带进位	SBCA	溢出置位	SEV
从累加器 B 中减去存储器数据，带进位	SBCB	从累加器 A 传入状态寄存器	TAP
从累加器 A 中减去存储器数据	SUBA	从状态寄存器传入累加器 A	TPA
从累加器 B 中减去存储器数据	SUBB	**其他操作：**	
从累加器 D 中减去存储器数据	SUBD	空操作	NOP
用二进制补码替换累加器 A 数据	NEGA	停止处理	STOP
用二进制补码替换累加器 B 数据	NEGB	专用测试模式	TEST
将累加器 A 中无符号数乘上累加器 B	MUL		

596 ~ 597

注：寄存器中位的个数取决于处理器。一个 8 位微处理器通常有 8 位的寄存器。有时，两个数据寄存器可一起使用，使位数加倍。这样的组合寄存器称为双寄存器。

Intel 8051

指令	助记符	指令	助记符
数据转移：		将累加器数据送到间接寻址外部 RAM 单元	MOVX @Ri，A
将立即数送到累加器 A 中	MOV A，#data	将累加器数据送到寄存器中	MOV Rn，A
将寄存器数据送到累加器 A 中	MOV A，Rn	将直接地址字节数据送到间接寻址 RAM 单元中	MOV @Ri，direct
将直接地址字节送到累加器中	MOV A，direct	将立即数送到寄存器中	MOV Rn，#data
将间接寻址 RAM 数据送给累加器	MOV A，@Ri	将直接地址单元字节数据送给直接地址单元	MOV direct，direct
将累加器数据送给直接地址字节	MOV direct，A		

（续）

指令	助记符	指令	助记符
将间接寻址寄存器 RAM 数据给直接地址单元	MOV direct，@Ri	长调用子程序	LCALL addr 16
将寄存器数据给直接地址单元	MOV direct，Rn	从子程序返回	RET
将立即数给直接地址单元	MOV direct，#data	从中断返回	RETI
将立即数给间接寻址单元 RAM	MOV @Ri，#data	**位操作：**	
16 位立即数送入数据指针寄存器	MOV DPTR，#data16	清除进位	CLR C
（A+DPTR）→A	MOVC A，@A+DPTR	指定位清零	CLR bit
将外部 RAM16 位地址中的数据送入累加器	MOVX A，@DPTR	将进位置位	SETB C
将累加器数据移入外部 RAM16 位地址中	MOVX @DPTR，A	指定位置位	SETB bit
交换累加器和直接地址单元数据	XCH A，direct	进位位 CY 取反	CPL C
交换累加器和间接寻址单元 RAM 数据	XCH A，@Ri	CY·bit→CY	ANL C，bit
交换寄存器和累加器数据	XCH A，Rn	CY·/bit→CY	ANL C，/bit
SP+1→SP，（direct）→（SP）	PUSH direct	CY+bit→CY	ORL C，bit
（SP）→（direct），SP-1→SP	POP direct	CY+/bit→CY	ORL C，/bit
跳转：		bit→CY	MOV C，bit
绝对跳转	AJMP addr 11	CY→bit	MOV bit，C
长跳转	LJMP addr 16	**逻辑操作：**	
短跳转，相对地址	SJMP rel	（direct）·A→（direct）	ANL direct，A
相对于 DPTR 间接跳转	JMP @A+DPTR	（direct）·data→（direct）	ANL direct，#data
如果累加器为零则跳转	JZ rel	A·data→A	ANL A，#data
累加器非零跳转	JNZ rel	A·（direct）→A	ANL A，direct
比较直接地址单元和累加器，不等则跳转	CJNE A，direct，rel	A·（Ri）→A	ANL A，@Ri
比较立即数和累加器，不等则跳转	CJNE A，#data，rel	A·Ri→A	ANL A，Rn
比较立即数和寄存器数据，不等则跳转	CJNE Rn，#data，rel	（direct）+A→（direct）	ORL direct，A
比较立即数和 Ri 间接寻址数据，不等则跳转	CJNE @Ri，#data，rel	（direct）+data→（direct）	ORL direct，#data
将寄存器减 1，非零则跳转	DJNZ Rn，rel	A+data→A	ORL A，#data
将直接地址单元减 1，非零则跳转	DJNZ direct，rel	A+（direct）→A	ORL A，direct
进位（CY=1）置位则跳转	JC rel	A+（Ri）→A	ORL A，@Ri
进位（CY=0）没有置位则跳转	JNC rel	A+Rn→A	ORL A，Rn
如果直接位为 1 则跳转	JB bit，rel	（direct）⊕A→（direct）	XRL direct，A
如果直接位为 0 则跳转	JNB bit，rel	（direct）⊕data→（direct）	XRL direct，#data
如果直接位为 1 转移，并使该位清零	JBC bit，rel	A⊕data→A	XRL A，#data
调用子程序：		A⊕（direct）→A	XRL A，direct
绝对调用子程序	ACALL addr 11	A⊕（Ri）→A	XRL A，@Ri
		A⊕Rn→A	XRL A，Rn
		加法：	
		将立即数加到累加器中	ADD A，#data
		将直接地址单元字节数据加到累加器中	ADD A，direct
		将间接寻址单元 RAM 数据加到累加器中	ADD A，@Ri
		将寄存器加到累加器中	ADD A，Rn
		将立即数加到累加器中，带进位	ADDC A，#data

（续）

指令	助记符	指令	助记符
将直接地址单元字节数据加到累加器中，带进位	ADDC A，direct	直接地址单元字节数据加 1	INC direct
		间接寻址单元 RAM 数据加 1	INC @ Ri
将间接寻址单元 RAM 数据加到累加器中，带进位	ADDC A，@ Ri	寄存器加 1	INC Rn
		累加器减 1	DEC A
将寄存器加到累加器中，带进位	ADDC A，Rn	直接地址单元数据减 1	DEC direct
减法：		间接寻址单元 RAM 数据减 1	DEC @ Ri
将立即数从累加器中减掉，带借位	SUBB A，#data	寄存器减 1	DEC Rn
		数据指针寄存器加 1	INC DPTR
将直接地址单元的数据从累加器中减掉，带借位	SUBB A，direct	**清零和取反操作：**	
		累加器按位取反	CPL A
将间接寻址单元 RAM 数据从累加器中减掉，带借位	SUBB A，@ Ri	累加器清零	CLR A
		循环移位操作：	
乘法和除法：		将累加器数据循环右移 1 位	RR A
将 A 和 B 相乘	MUL AB	将累加器数据带进位循环右移 1 位	RRC A
将 A 除以 B	DIV AB		
十进制运算：		将累加器数据循环左移 1 位	RL A
A0 ~ 3←→（Ri）0 ~ 3	XCHD A，@ Ri	将累加器数据带进位循环左移 1 位	RLC A
累加器内半字节交换	SWAP A		
将累加器数据进行十进制调整	DA A	**空操作：**	
递增和递减：			
累加器加 1	INC A	空操作	NOP

注：“·”是与运算符，“+”是或运算符，“/”是非运算符，“⊕”是异或运算符。
前面有标号#代表 1 个数字，#data16 表示 1 个 16 位常量；Rn 表示寄存器的内容，@ Ri 表示寄存器指针指向的地址里的数据；DPTR 是 1 个数据指针；direct 是 1 个存储器地址，在地址里可以找到指令所使用的数据。

598 ~ 599

PIC16Cxx

指令	助记符	指令	助记符
w 和一个数相加	addlw number	f 内容加 1	Incf FileReg，f
w 和 f 相加，并将数据存储在 f 中	addwf FileReg，f	w 和立即数逻辑或	iorlw number
w 和 f 相加，并将数据储存在 w 中	addwf FileReg，w	w 和 f 逻辑或	iorwf FileReg，f
w 和一个数逻辑与，并将结果存储在 w 中	andlw number	把 f 中的数传送到 w	movf FileReg，w
		把 w 内容传送到 f	movwf FileReg
w 和 f 逻辑与	andwf FileReg，f	将数值存放于 w 内	movlw
f 的位清 0	bcf FileReg，bit	空操作	nop
f 的位置 1	bsf FileReg，bit	中断返回	retfie
测试 f 的位，为 0 则跳过下一条指令	btfsc FileReg，bit	将立即数存放在 w 内，从子程序返回	retlw number
测试 f 的位，为 1 则跳过下一条指令	btfss FileReg，bit	从子程序返回	return
		带进位 f 循环左移	rlf FileReg，f
调用子程序，子程序执行完后，返回到离开的位置	call AnySub	带进位 f 循环右移	rrf FileReg，f
		使 PIC 进入睡眠，省电模式	sleep
f 清零	clrf FileReg	立即数减去 w	sublw number
w 清零	clrw	f 减去 w	subwf FileReg，f
看门狗定时器清零	clrwdt	f 高低 4 位内容交换	swap FileReg，f
f 取反	comf FileReg，f	用 w 中的数指定哪些位是输出或输入	tris PORTX
f 减 1	decf FileReg，f		
f 减 1，结果为 0 则跳过下一条指令	decfsz FileReg，f	w 和立即数逻辑异或	xorlw number
无条件转移到 label	goto label	w 和 f 逻辑异或	xorwf FileReg，f

注：f 用于文件寄存器，w 用于工作寄存器，b 用于指位。助记符表示了有关的操作数的类型，例如 movlw 表示用 w 进行 move 操作（将一个数值存入 w 中）；movwf 表示用 w 寄存器和 f 寄存器进行 move 操作。

600

附录E　C库函数

下面是一些通用的C库函数，但并没有列出任何一个库函数的全部内容，也没有列出所有的库。任何一个编译器都可以得到这些库。

< ctype. h >

isalnum	int isalnum(int ch)	检测字母数字字符，如果参数是字母或者数字则返回非0，否则返回0。
isalpha	int isalpha(int ch)	检测字母字符，如果是字母返回非0，否则返回0。
iscntrl	int iscntrl(int ch)	检测控制字符，如果是0～0x1F或者等于0x7F（DEL），则返回非0，否则返回0。
isdigit	int isdigit(int ch)	检测十进制数字字符，如果是数字0～9，则返回非0，否则返回0。
isgraph	int isgraph(int ch)	检测可打印字符（除了空格），如果是可打印的，则返回非0，否则返回0。
islower	int islower(int ch)	检测小写字母字符，如果是小写字母，则返回非0，否则返回0。
isprint	int isprint(int ch)	检测可打印字符（包括空格），如果是可打印的，则返回非0，否则返回0。
ispunct	int ispunct(int ch)	检测标点符号字符，如果是标点符号字符，则返回非0，否则返回0。
isspace	int isspace(int ch)	检测空格字符，如果是空格符、制表符、换页符、回车、换行符，则返回非0，否则返回0。
isupper	int isupper(int ch)	检测大写字母字符，如果是大写字母，则返回非0，否则返回0。
isxdigit	int isxdigit(int ch)	检测十六进制字符，如果是十六进制数字，则返回非0，否则返回0。

< math. h >

acos	double acos(double arg)	返回参数的反余弦值。
asin	double asin(double arg)	返回参数的反正弦值。
atan	double atan(double arg)	返回参数的反正切值。只需要一个参数。
atan2	double atan2(double y,double x)	返回y/x的反正切值。
ceil	double ceil(double num)	返回大于等于num的最小整数。
cos	double cos(double arg)	返回arg的余弦值，参数值必须用弧度表示。
cosh	double cosh(double arg)	返回arg的双曲余弦值。

exp	double exp(double arg)	返回 e^x 的值，其中 x 是参数。
fabs	double fabs(double num)	返回 num 的绝对值。
floor	double floor(double num)	返回小于等于 num 的最大整数。
fmod	double fmod(double x,double y)	返回 x/y 的浮点形式的余数。
idexp	double idexp(double x,int y)	返回 x 乘以 2^y。
log	double log(double num)	返回 num 的自然对数。
log10	double log10(double num)	返回 num 的以 10 为底的对数。
pow	double pow(double base,double exp)	返回以参数 base 为底的 exp 次幂。
sin	double sin(double arg)	返回 arg 的正弦值。
sinh	double sinh(double arg)	返回 arg 的双曲正弦值。
sqrt	double sqrt(double num)	返回 num 的平方根。
tan	double tan(double arg)	返回 arg 的正切值。
tanh	double tanh(double arg)	返回 arg 的双曲正切值。

<stdio. h>

getchar	int getchar(void)	返回在键盘上输入的下一个字符。
gets	char gets(char *str)	读取键盘上的输入字符直到读到回车符。将这些字符存取并放入字符串 str。
printf	int printf(char *str, …)	输出字符串 str 中的字符。
puts	int puts(char *str)	输出字符串 str 中的字符。
scanf	int scanf(char *str, …)	从键盘读取输入的信息放入字符串。

<stdlib. h>

abort	void abort(void)	程序立即终止。
abs	int abs(int num)	返回整数 num 的绝对值。
bsearch	void bsearch (const void * key, const void *base,size_t num,size_t size,int (*compare)(constvoid *, const void *))	对 base 指向的数组进行二分法搜索，并返回一个指针，指向与 key 相匹配的第一个元素，数组元素的个数存在 num 中，每个元素的大小（以字节为单位）存在 size 中。
calloc	void * calloc (size _t num, size _t size)	在内存的动态存储区中分配一个长度为 num、元素大小为 size 的数组空间，函数返回一个指向分配起始地址的指针。
exit	void exit(int status)	立即正常停止程序，状态值传送到调用程序。
free	void free(void *ptr)	释放 ptr 指向的分配内存。
labs	long labs(long num)	返回长整型数 num 的绝对值。
malloc	void *malloc(size_t size)	返回一个指针，该指针指向向系统申请分配 size 个字节的内存空间的第一个字节。
qsort	void qsort(void *base, size_t num, size_t size,int(*compare)(const void *,const void *))	将 base 指向的数组排序，num 是这个数组元素的个数，size 是一个元素的大小。
realloc	void * realloc (void * ptr, size _t size)	改变 ptr 指向的内存大小，其大小由 size 指定。

注：size_t 是“size of”变量的类型定义，通常代表另一个对象或者参数的大小。

<time. h>

asctime	char * asctime(const struct tm *ptr)	将时间从 structure 型变为字符串形式以适于显示，并返回指向该字符串的指针。
clock	clock_t clock(void)	返回程序开始之后运行的时钟周期数。
ctime	char *ctime(const time_t *time)	把日期和时间转换为字符串，形式为 day month date hours：minutes：seconds year \n \0，它是由指向秒数的指针给定，秒数是从格林尼治标准时间 00：00：00 开始计时的。
difftime	double difftime (time _ ttime2, time _ t time1)	用秒为单位返回 time1 和 time2 的差值。
gmtime	struct tm *gmtime(const time_t *time)	将时间由长整型转换为结构型，并返回其指针。
localtime	struct tm *loclatime(const time_t *time)	将当地时间由长整型转换为结构型，并返回其指针。
time	time_t time(time_t *system)	返回当前系统的日历时间。

注：time_t 和 clock_t 用作 “time of” 和 “number of cycles of” 的变量类型。

附录 F　MATLAB 和 SIMULINK

F.1　MATLAB

可用于辅助计算和系统建模的计算机软件，其中经常使用的一个程序就是 MATLAB。下面是对 MATLAB（Mathworks Inc. 的注册商标）的简要介绍。想要进一步了解 MATLAB，读者可以参考用户指南或者一些教科书，如 Hahn，B. 编著的《Essential MATLAB for Engineers and Scientists》第 5 版（Elsevier 2012）或 Moore，H. 编著的《MATLAB for Engineers》（Pearson 2013）。

在提示符的后面输入命令，然后按回车（ENTER 或 return）键执行命令。在下面的命令介绍中，不需要重复按回车键，而且假定在其他情况下也应该如此。在 Windows 或者 Macintosh 系统中启动 MATLAB，可以通过单击 MATLAB 图标或者输入 MATLAB。计算机屏幕会显示出 MATLAB 提示符。关闭 MATLAB 则需要在提示符后输入 exit 或者 quit。因为 MATLAB 是要区分大小写的，所以命令中用的是小写字母。

在提示符后输入 help 或者点击 MATLAB 窗口上方菜单栏中的帮助（help）选项，会弹出一个 MATLAB帮助主题列表。为了从列表中获取某个特殊主题的帮助，例如指数（exponential），可以输入 help exp。输入 lookfor 加某些主题，将引导 MATLAB 寻找该主题的相关信息。比如输入 lookfor integ，可以显示关于积分（integration）的一些命令。

通常，输入 MATLAB 的数学操作和在纸上的写法是一样的。例如：

```
≫ a = 4/2
```

得到：

```
a =
    2
```

再如：

```
≫ a = 3 * 2
```

得到：

```
a =
    6
```


MATLAB 中运算符的运算顺序为：指数运算（^）、乘（*）、除（/）、加（+）、减（-）。其余按照从左到右，且有括号的先计算括号里的。例如：

```
≫ a = 1 + 2^3/4 * 5
```

得到：

```
a =
    11
```

若计算 1 + 2^3/(4 * 5)，在提示符后输入：

```
≫ a = 1 + 2^3/(4 * 5)
```

则得到：

```
a =
    1.4
```

因为这里是用2^3来除以 4 和 5 的乘积，所以结果为 1.4。

下面是 MATLAB 的一些数学函数：

abs(x)	x 的绝对值，即 $\|x\|$
exp(x)	x 的指数，即 e^x
log(x)	x 的自然对数，即 $\ln(x)$
log10(x)	以 10 为底的 x 的对数，即 $\log_{10}(x)$
sqrt(x)	x 的平方根，即$\sqrt{x}$
sin(x)	弧度 x 的正弦运算 $\sin(x)$
cos(x)	弧度 x 的余弦运算 $\cos(x)$
tan(x)	弧度 x 的正切运算 $\tan(x)$
asin(x)	反正弦运算，即 $\sin^{-1} x$
acos(x)	反余弦运算，即 $\cos^{-1} x$
atan(x)	反正切运算，即 $\tan^{-1} x$
csc(x)	计算 $1/\sin x$
sec(x)	计算 $1/\cos x$
cot(x)	计算 $1/\tan x$

π 的输入方法是：pi。

此外，除了将一系列命令写在提示符后外，还可以将命令写在一个文本文件中，运行命令时只需将 MATLAB 指向该文件即可。这里用到的文本文件叫作 M 文件，里面包含了连续的 MATLAB命令，该文件以 .m 为后缀。编辑 M 文件时，应以 function 开头，后面依次是函数名，输入和输出，如下所示：

```
Function [output] = function name [input]
```

例如，function y = cotan(x)，这就是用 cotan(x) 确定 y 值的文件。这个文件可以在一系列 MATLAB 命令中被调用，调用格式为函数名和输入变量，例如 cotan(x)。事实上，MATLAB中已包含了这个文件，且当 x 的余切值需要时就可使用。一个多输入参数的函数需要在声明的时候列出所有的输入参数。类似地，一个多返回值的函数也需要列出所有的输出。

605

以% 开头的行是注释行。它们不会被 MATLAB 解释为命令。例如，我们需要设计程序来求一列数据点的方均根值，可以这样写程序：

```
function y = rms(x)
% rms 是方均根值
% rms(x) 给出一个矢量 x 的元素的方均根值
xs = x.^2;
s = size(x);
y = sqrt(sum(xs)/s(2));
```

我们可令 xs 为每个 x 的平方值。命令 s = size(x) 求取括号中输入数据的个数，命令 y = sqrt(sum(xs)/s(2)) 求所有 xs 的和除以 s 的平方根。符号“;”用于每句命令的句尾。

MATLAB 提供多个包含 M 文件的工具箱，与本书特别相关的是控制系统工具箱（Control System Toolbox）。可用于执行脉冲、阶跃、斜坡的时间响应，并有伯德图和奈奎斯特图分析、

根轨迹分析等。例如，为了生成一个传递函数为$\frac{4}{s^2+2s+3}$的系统的伯德图，可以这样编程：

```
%生成 G(s) = 4/(s^2 +2s +3) 的伯德图
num = 4
den = [1 2 3];
bode (num, den)
```

bode(num,den）命令产生增益（dB 为单位）和相位（度数为单位）分别对频率（单位是 rad/s）的 Bode 图曲线，横坐标为频率的对数刻度。

F.1.1　绘图

二维线性曲线图可以使用命令 plot(x,y）生成，画出 x 和 y 的值，例如：

```
x = [0 1 2 3 4 5];
y = [0 1 4 9 16 25];
plot(x, y)
```

为了绘制一个函数的曲线，无论是标准的还是用户定义的，都使用命令 fplot(function name, lim）命令，其中 lim 确定了绘制的间隔，即 x 的最大或最小值。

命令 semilogx(x,y）生成 x 和 y 的图像，x 是对数标度，y 为线性标度。命令 semilogy(x,y）生成 x 和 y 的图像，y 是对数刻度，x 为线性刻度。命令 loglog(x,y）生成的图像中 x 和 y 均使用对数刻度。命令 polar(theta,r）生成极坐标下的图像，theta 是弧度下的变量，r 是幅值。

606

subplot 命令使图形窗口分为多个子窗口，并在每个窗口分别绘图。例如，

```
x = [0 1 2 3 4 5 6 7];
y = exp(x);
subplot(2,1,1);plot(x,y);
subplot(2,1,2);semilogy(x,y);
```

subplot 命令给出了 3 个整型量 m、n、p，表示把窗口分为 $m \times n$ 个子窗口，其中 m 为行数，n 为列数，p 指定了图像位于哪个子窗口。小窗口按照从左到右、从上到下的顺序编号。因此上面的例子就是把窗口分为两行一列，每行一个图像。上面的是线性曲线图形，下面的是半对数图形。

还可以选择网格线的数目和类型、图形颜色和文字说明。print 命令用于将图形输出到文件或者打印机。这可以通过图像窗口中选择文件菜单项，再选择打印选项来实现。

F.1.2　传递函数

下面举例说明怎样在屏幕上输入并显示传递函数：

```
%G(s) = 4(s +10)/(s +5)(s +15)
num = 4*[1 10];
den = conv([1 5],[1 15]);
printsys(num,den,'s')
```

num 命令用来表示传递函数的分子的系数，按照 s 的降幂排列。den 命令表示分母中的每个多项式的系数，也是按照 s 的降幂排列。conv 表示两个多项式相乘，在上面例子中就是（$s+5$）和($s+15$)。printsys 命令表示在 s 域内以分数形式在屏幕上显示传递函数。

有时我们需要根据传递函数中的两个多项式的比来确定极点和零点。例如：

```
%求传递函数的零点和极点
%G(s) = (5s^2 +3s +4)/(s^3 +2s^2 +4s +7)
num = [5 3 4];
```

```
den = [1 2 4 7];
[z, p, k] = tf2zp (num, den)
```

[z, p, k] = tf2zp(num,den) 是用于确定并显示出输入的传递函数的 zeros(z)、poles(p) 和 gain(k)。

MATLAB 可以显示曲线表示系统在不同输入下的响应。例如，下面的程序可以给出系统在单位阶跃输入下对给定传递函数的响应：

607

```
%显示单位阶跃输入下系统的响应
%传递函数为 G(s) = 5/(s^2 +3s +12)
num = 5;
den = [1 3 12];
step(num,den)
```

F.1.3　框图

控制系统经常用一系列相互连接的框图来表示，每一个方框都有其指定的特征。MATLAB 允许用相互连接的框图来建立系统。当一个给定的开环传递函数的方框有一个单位反馈以后，使用 cloop 命令。如果反馈不是单位的，可用 feedback 命令，如图 F-1 所示，我们有这样程序：

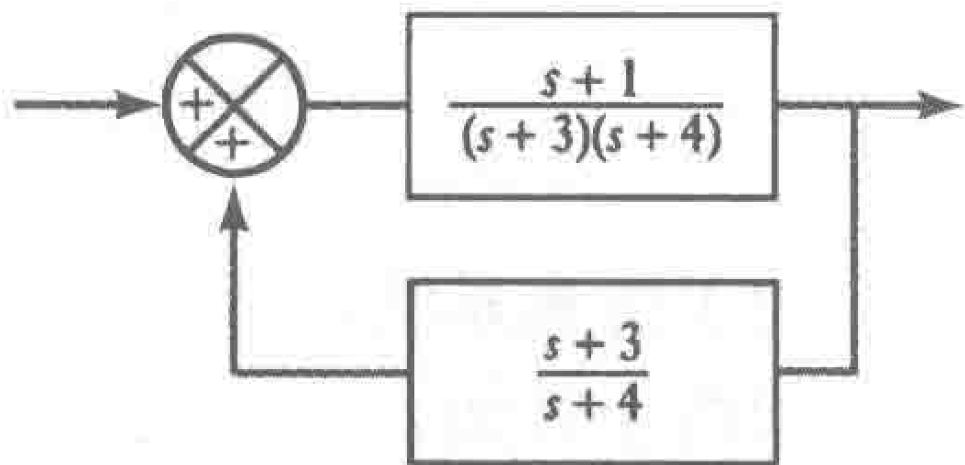


图 F-1　框图

```
%带有反馈环的系统
ngo = [1 1];
dgo = conv([1 3],[1 4]);
nh = [1 3];
dh = [1 4];
[ngc2, dgc2] = feedback (ngo, dgo, nh, dh)
printsys (ngc2,dgc2,'s')
```

其中 ngo 和 dgo 分别是开环传递函数 $G_0(s)$ 的分子和分母，nh 和 dh 分别是反馈传递函数 $H(s)$ 的分子和分母。该程序将以一个整体的形式显示出这个控制系统。

series 命令用于表示两个方框在某个路径下串联起来，parallel 命令用来表示它们是并联的。

F.2　SIMULINK

SIMULINK 和 MATLAB 一起使用，可以通过连接屏幕上的方框来构建系统，而不是用前面介绍的通过一系列 MATLAB 命令构建框图来构建。MATLAB 启动后，可以使用≫SIMULINK 来启动控制窗口以及其中的图标和标题栏中的下拉菜单。单击 file 和下拉菜单中的 new，可以打开一个窗口并在该窗口内组装系统（见图 F-2）。

为了创建需要的方框，回到控制窗口双击线性图标。单击并拖动传递函数图标至未命名窗口。如果想得到增益方框，单击并拖动增益图标至未命名窗口。同样的操作可用于求和图标和积分图标。这样就可以把所有需要的方框拖进未命名窗口。然后双击 Sources 图标，在下拉菜单中选择合适的输入，比如阶跃输入，然后拖进未命名窗口。现在双击 sinks 图标，也把图标拖进来。当鼠标指针位于某个图标的输出端时，按住鼠标，拖动到下一个图标的输入端，就可以把两个块连接起来。重复上述操作直至所有部件都连接起来。

608

为了给出一个传递函数，双击 transfer Fcn 方框，弹出对话框，使用 MATLAB 命令输入分子分母。例如，如果分子是 $s+1$，就单击分子，输入 [1 1]，然后单击 done 图标。如果分母是 (s^2+2s+3)，单击分母，输入 [1 2 3]，然后单击 done 图标。双击 gain 图标，输入增益值。双击 sum 图标，根据正反馈还是负反馈选择是 + 还是 -。双击 graph 图标，设置图形参数。

现在，完整的仿真系统就显示在屏幕上。图 F-2 就是可能出现的完整系统示意图。如果要删除某个块或者连接，选中并单击，然后按 < DEL > 键。

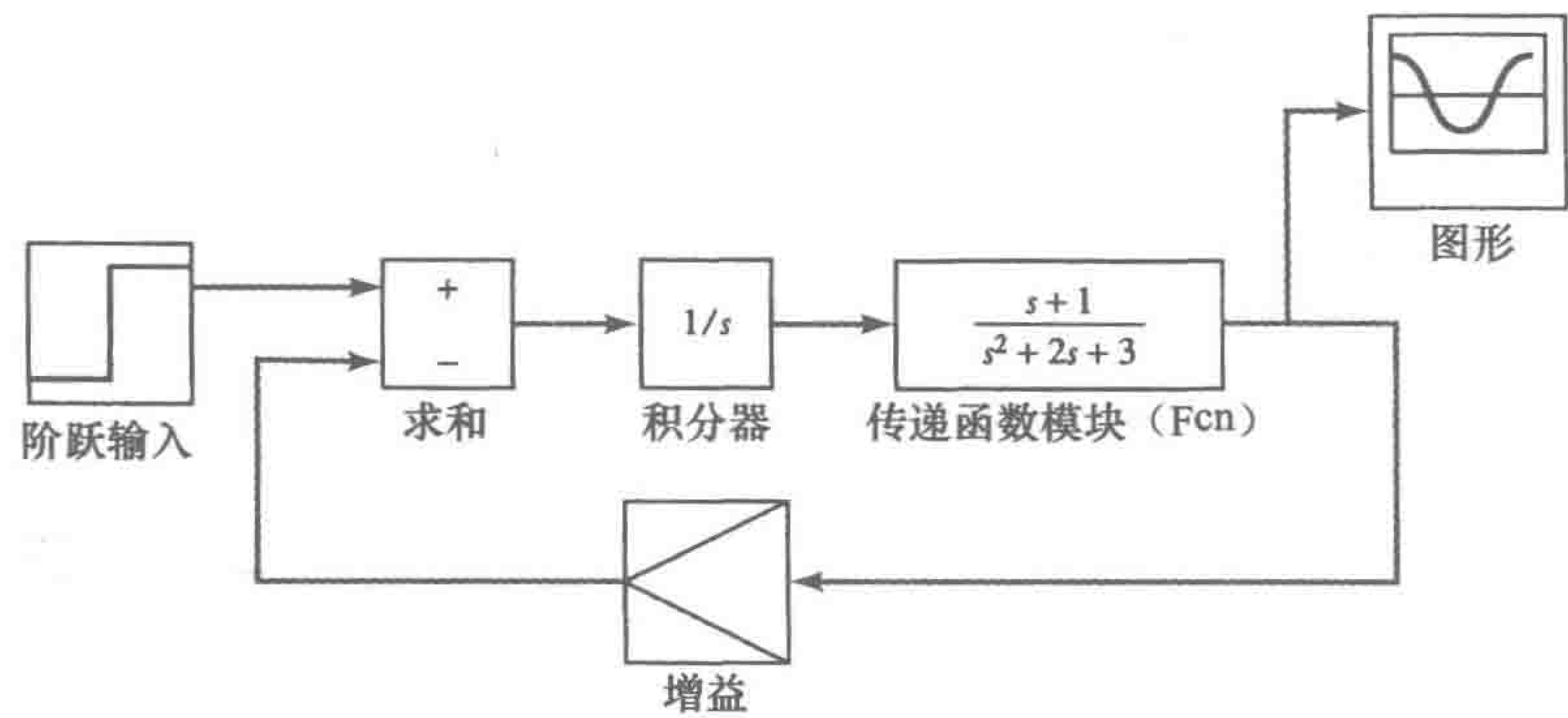


图 F-2　SIMULINK 的使用实例

为了仿真系统的行为，单击 Simulation 按钮打开下拉菜单。选择 Parameters，并设置仿真的开始和终止时间。在 Simulation 菜单下，选择 Start。然后 SIMULINK 就会建立图形窗口，并显示系统输出。然后选择 File，并在下拉菜单中单击 SAVE AS，可以保存文件。在弹出的对话框中输入一个文件名，然后点击 Done 即可。 609

附录G 电路分析

G.1 直流电路

电路分析中的基础定律是基尔霍夫定律（Kirchhoff's law）：

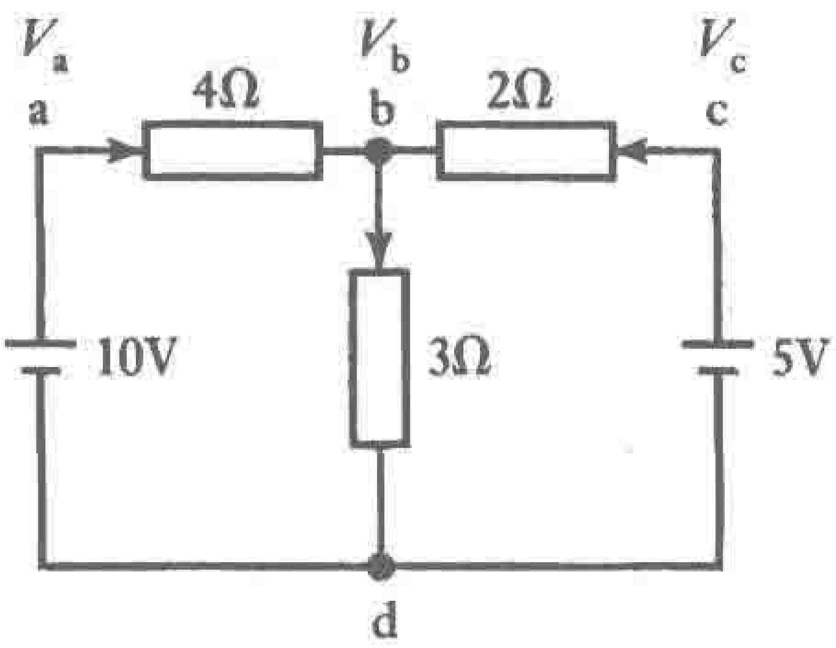
1）**基尔霍夫电流定律**（Kirchhoff's current law）表明，对于电路中的任意节点，流进的电流等于流出的电流。

2）**基尔霍夫电压定律**（Kirchhoff's voltage law）表明，环绕电路中的任意闭合环路，经过所有元件电压降落之和等于电压升高之和。

通常，通过系统地确定等效电阻，可将一个有串并联电阻组合的电路简化为简单电路，从而简化为简单电路的分析问题。对于更复杂的电路，可能会用到以下方法。

G.1.1 节点分析

节点（node）是指电路中两个或两个以上支路的连接点，即节点是电流流进和电流流出的连接点。**主要节点**是指三个或更多元件的连接点。在图G-1中，只有b和d是主要节点。其中一个主要节点作为参考节点，以便以它为参考得到另一节点的电势差。对图G-1进行分析，d作为参考节点，对其他非参考节点应用基尔霍夫电流定律。步骤如下：



图G-1 节点分析法的电路

1）画出带有标记的电路图并标识出主要节点。

2）选择一个主要节点作为参考节点。

3）对每个非参考节点应用基尔霍夫电流定律，利用欧姆定律根据节点电压表示通过电阻的电流。

4）联立方程组并求解。如果有 n 个主要节点就会有 $n-1$ 个方程。

5）利用求得的节点电压值确定电路中每一支路的电流。

以图G-1为例，电路中有节点a、b、c、d，其中b和d是主要节点，节点d作为参考节点。

610

如果 V_a、V_b 和 V_c 是相对于节点d的节点电压，则通过4Ω电阻的电势差是 V_a-V_b，通过3Ω电阻的是 V_b，通过2Ω电阻的为 V_c-V_b。因此流过4Ω、3Ω和2Ω电阻的电流分别为（V_a-V_b）/4、V_b/3和（V_c-V_b）/2。对节点b应用基尔霍夫电流定律，有：

$$\frac{V_a-V_b}{4}+\frac{V_c-V_b}{2}=\frac{V_b}{3}$$

因为 $V_a=10V$，$V_c=5V$，所以

$$\frac{10-V_b}{4}+\frac{5-V_b}{2}=\frac{V_b}{3}$$

可以得到 $V_b=4.62V$。4Ω电阻两侧的电势差为 $10-4.62=5.38V$，流过的电流为 $5.38/4=1.35A$。同理，对于3Ω电阻，电势差为4.62V，电流为 $4.62/3=1.54A$；对于2Ω电阻，电势差为 $5-4.62=0.38V$，电流为 $0.38/2=0.19A$。

G.1.2 网孔分析法

回路（loop）是指一系列的电路元件所形成的封闭路径。**网孔**（mesh）是指不包含任何其

他回路的一个回路。网孔分析法需要为每个网孔定义一个环绕电流。每个网孔电流需要有相同的方向，通常选择顺时针方向作为所有网孔电流的流动方向。确定网孔电流以后，对每个网孔应用基尔霍夫电压定律，步骤如下：

1）为每个网孔标出顺时针方向的网孔电流。

2）对每一个网孔应用基尔霍夫电压定律，根据欧姆定律，用通过每个电阻的电流表示出每个电阻的电势差，电势差方向与电流方向相反。只与一个网孔邻接的电阻，通过它的电流即为网孔电流；与两个网孔邻接的电阻，通过它的电流为两个网孔电流的代数和。

3）联立方程组，求解得到网孔电流，若有 n 个网孔则有 n 个方程。

4）利用求得的网孔电流确定每个支路的电流。

以图 G-2 中的电路图为例，图中有 3 个回路（ABCF、CDEF 和 ABCDEF），但其中只有前两个是网孔。我们定义 I_1 和 I_2 分别为网孔内的顺时针电流。

611

对于网孔 1，应用基尔霍夫电压定律有 $5 - 5I_1 - 20(I_1 - I_2) = 0$，即

$$5 = 25I_1 - 20I_2$$

同理，对于网孔 2，有 $-10I_2 - 20 - 20(I_2 - I_1) = 0$，也即

$$20 = 20I_1 - 30I_2$$

由以上两个方程可以得到 $I_2 = -1.14\text{A}$，$I_1 = -0.71\text{A}$。负号表示电流方向与图中所示的方向相反。通过 20 Ω 电阻的电流与 I_1 同向且为 $-0.71 + 1.14 = 0.43\text{A}$。

G.1.3　戴维南定理

包含一个电压源或者电流源的二端网络，其等效电路可通过**戴维南定理**（Thévenin's theorem）得到。定理内容如下：

> 任何包含电压源或电流源的二端网络（见图 G-3a）可以由包含一个电压源和一个与之串联的阻抗所构成的电路来代替。此电压源电压为初始电路的开路电压，而串联阻抗是在两端点间无负载且网络里所有独立源为零（见图 G-3b）的情况下测得的。

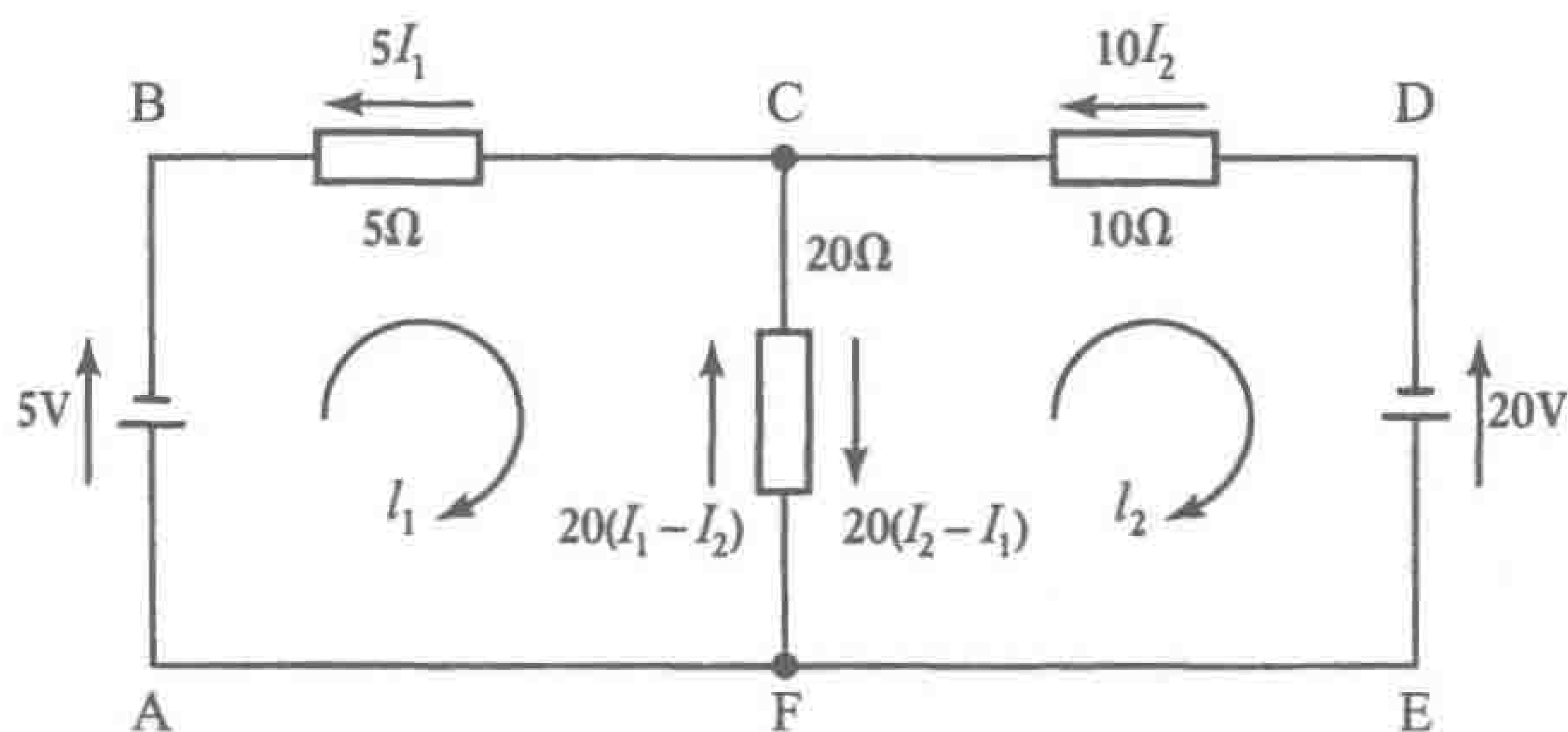


图 G-2　网孔分析法的电路

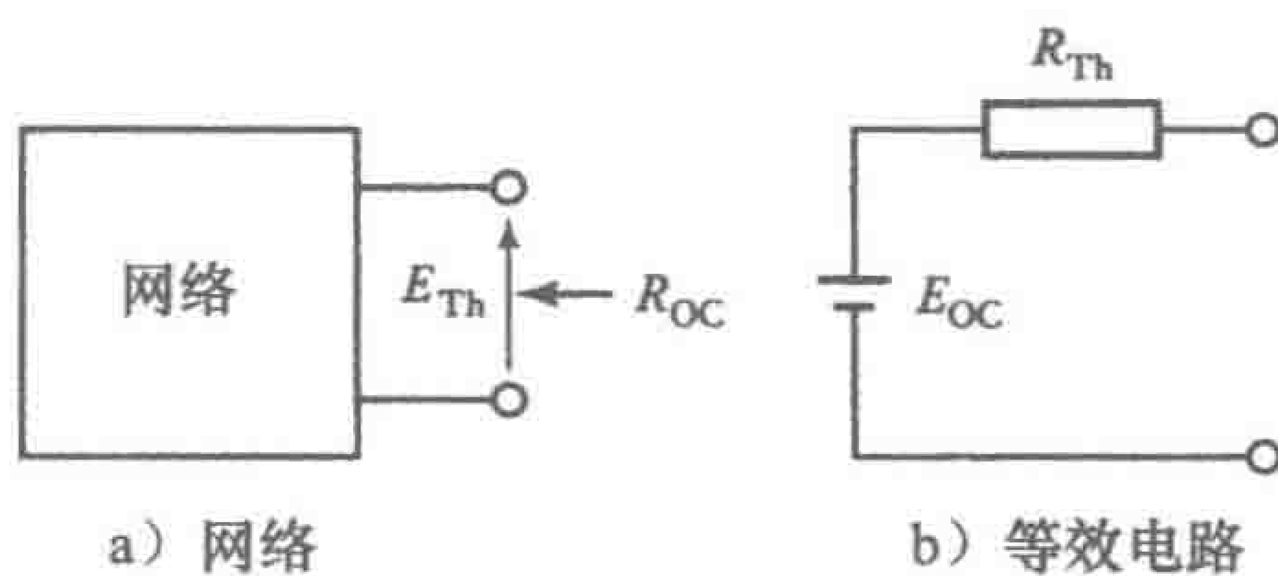


图 G-3　网络与等效电路

如果电路为线性电路，为了应用戴维南定理，需要将电路分为两个电路 A 和 B，两个电路在一对端点处相连。利用戴维南定理，用等效电路替换电路 A。当电路 B 未连接时，电路 A 的开路戴维南电压可以求得。电路 A 的戴维南阻抗是在将所有独立源为 0 时，从 A 的端口看进去的阻抗。图 G-4 说明了整个步骤。

612

1）确定电路的 A 和 B 两部分，并用端点将其分开。

2）将 A 部分与 B 部分分开。

3）将 A 用其戴维南等效电路代替，即一个电压源和一个串联阻抗。

4）重新将其与 B 部分连接并进行分析。

举例说明利用戴维南定理求解图 G-5 电路中 10Ω 电阻流过的电流。

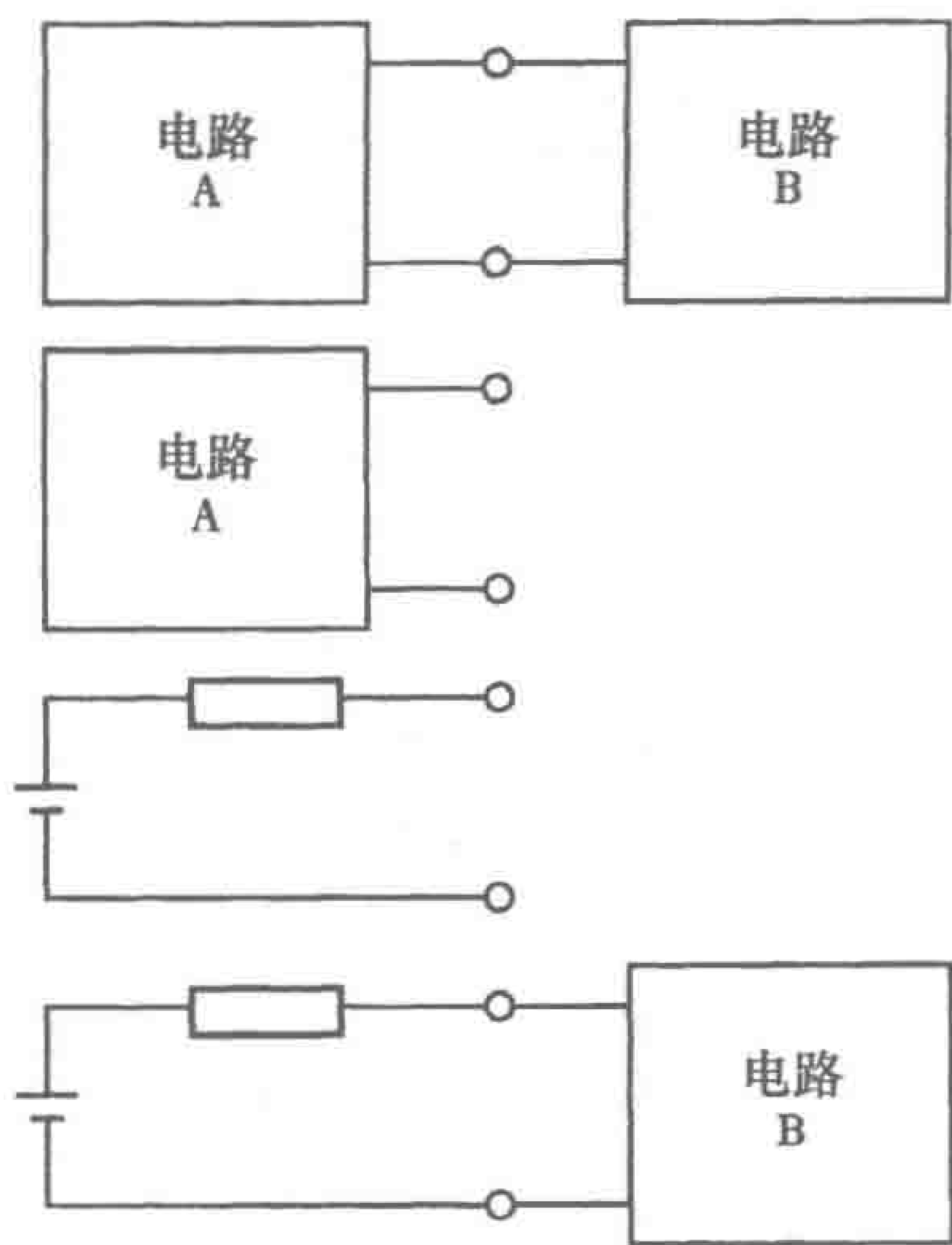


图 G-4　电路分析步骤

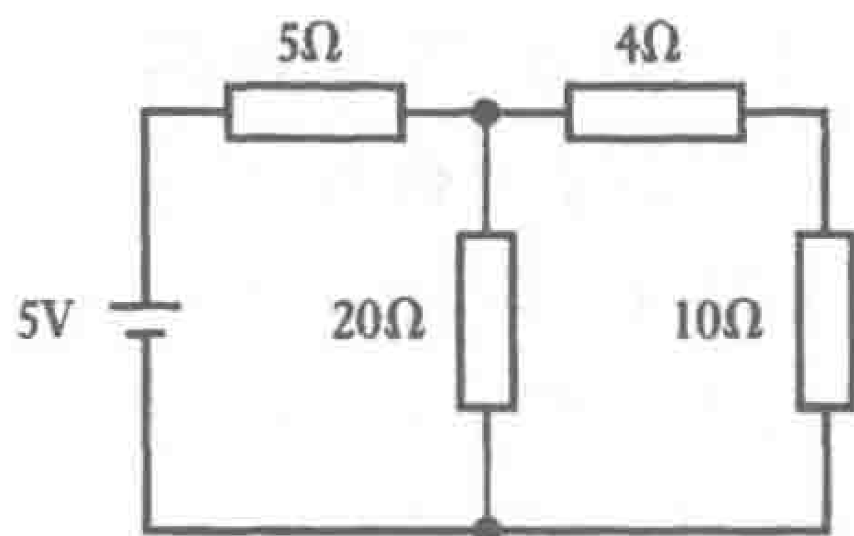


图 G-5　应用戴维南定理的示例电路

因为我们只对 10Ω 电阻流过的电流感兴趣，所以将其作为 B 网络，电路的剩余部分均作为 A 网络，两者由端点连接（见图 G-6a）。将 A、B 分开（见图 G-6b）并求其戴维南等效电路。

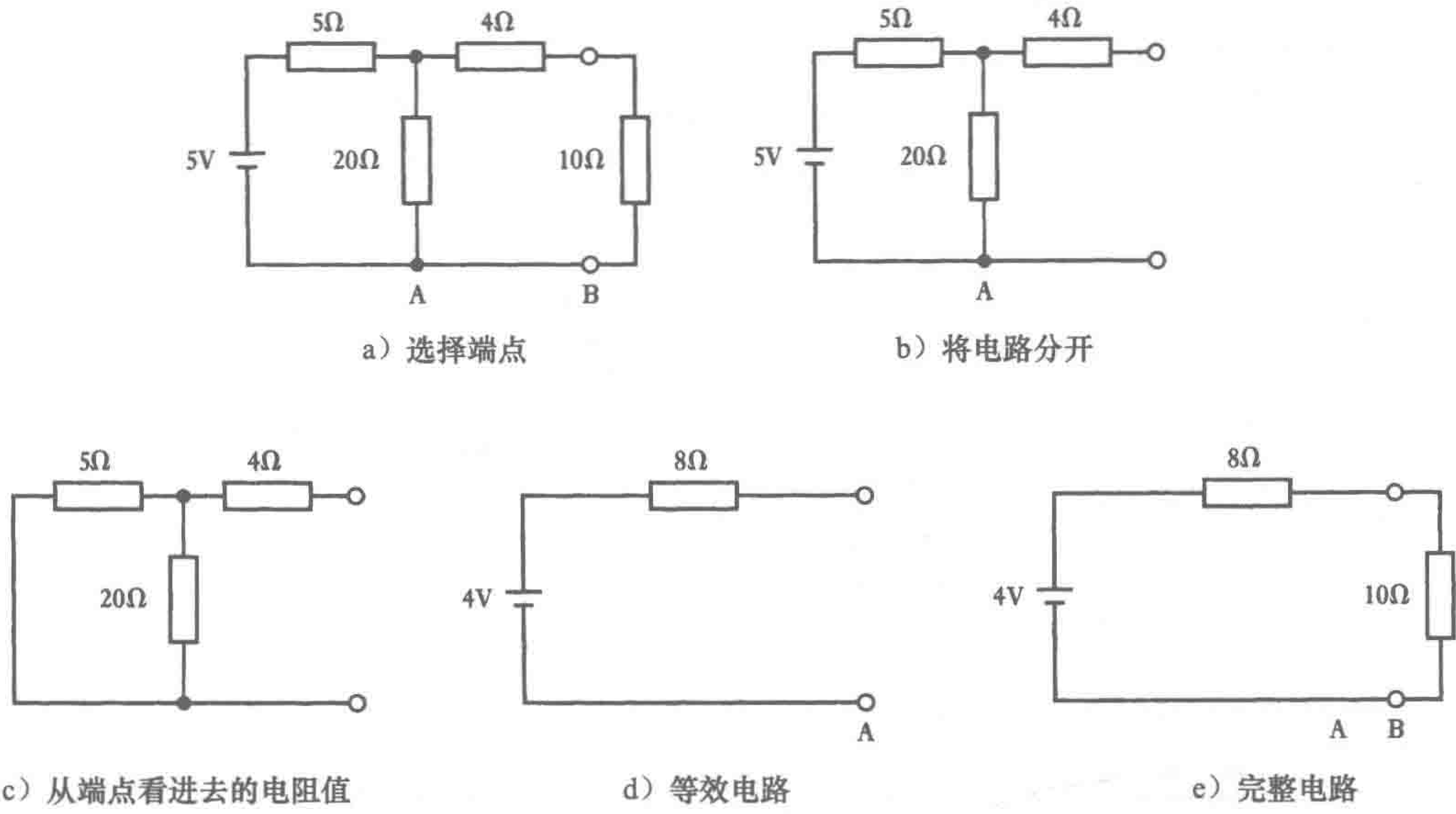


图 G-6　戴维南分析

开路电压为 20Ω 电阻的两端电压，即总压降中通过 20Ω 所占的部分为：

$$E_{Th} = 5\frac{20}{20+5} = 4V$$

当电压源为 0 时，从端点看进去的电阻值是 5Ω 电阻与 20Ω 电阻并联以后再与 4Ω 电阻串联得到的电阻值（见图 G-6c）：

$$R_{Th} = 4 + \frac{20 \times 5}{20+5} = 8\Omega$$

所以戴维南等效电路如图 G-6d 所示，当网络 B 与其连接后得到的电路如图 G-6e 所示。所以通过 10Ω 电阻的电流为 $I_{10} = 4/(8+10) = 0.22A$。

G.1.4　诺顿定理

与戴维南定理相似，对于任意包含电压源或者电流源的二端网络，可以将电路等效为一个电流源与一个阻抗并联，这就是**诺顿定理**（Norton's theorem）：

> 含有电压源或电流源的任意二端网络可以用一个电流源与一个并联阻抗组成的等效电路来代替。等效电流源电流为流过端点的短路电流，等效阻抗为两端点间无负载且独立源为零时，两端点间测得的阻抗值。

对于线性电路，需要将其分为由一对端点连接的两部分电路 A 和 B（见图 G-7）。然后利用诺顿定理将电路 A 用其等效电路代替。当电路 B 未连接时，电路 A 的诺顿短路电流可以求得。当所有独立源为零时，电路 A 的诺顿阻抗即为从 A 的端点看进去的阻抗值。

1）确定电路的 A 和 B 两部分，并用端点将其分开。

2）将 A 部分与 B 部分分开。

3）将 A 用其诺顿等效电路代替。

4）重新将其与 B 部分连接并进行分析。

举例说明利用诺顿定理求解图 G-8 中流过 20Ω 电阻的电流 I。

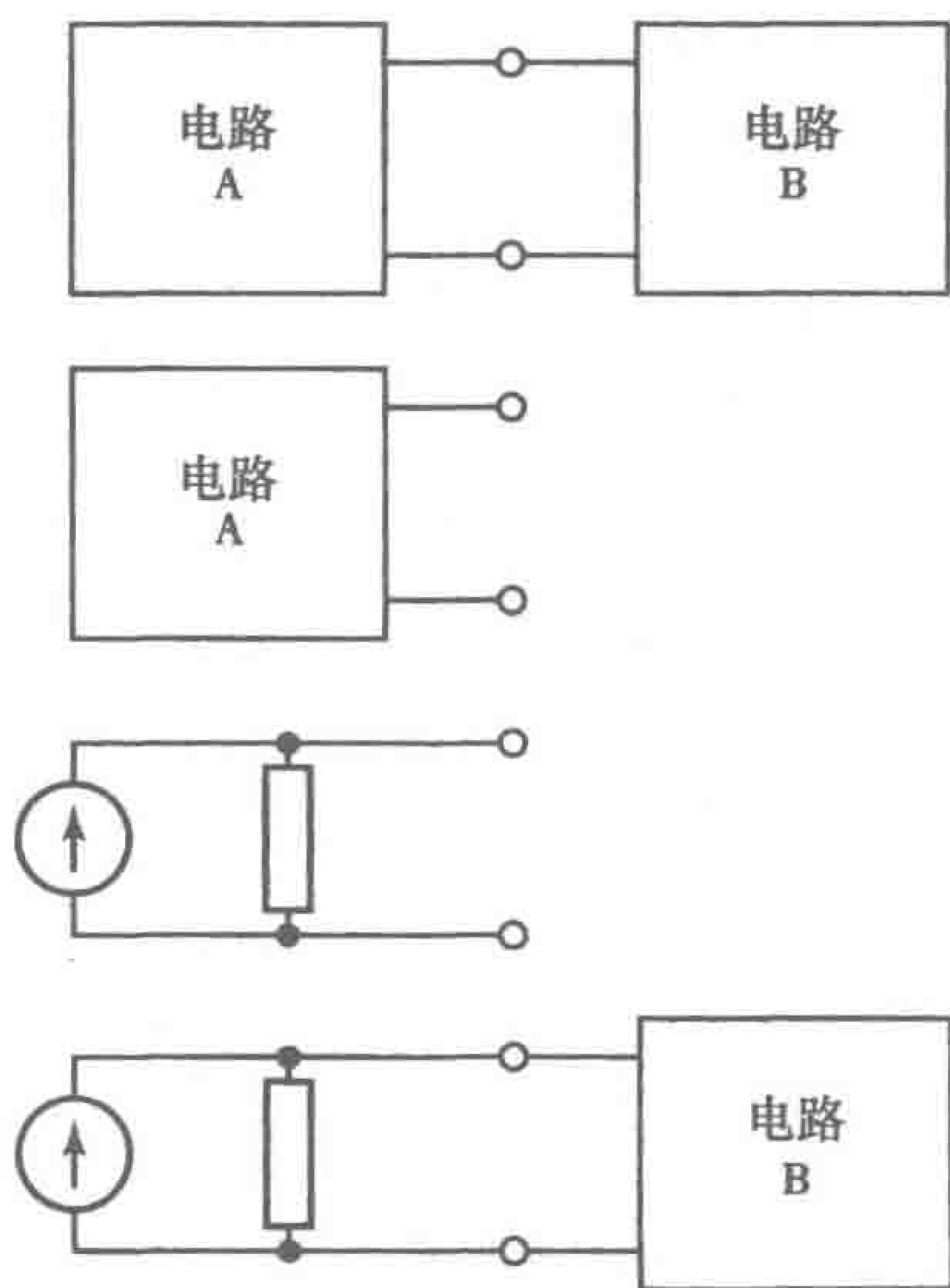


图 G-7　应用诺顿定理分析电路步骤

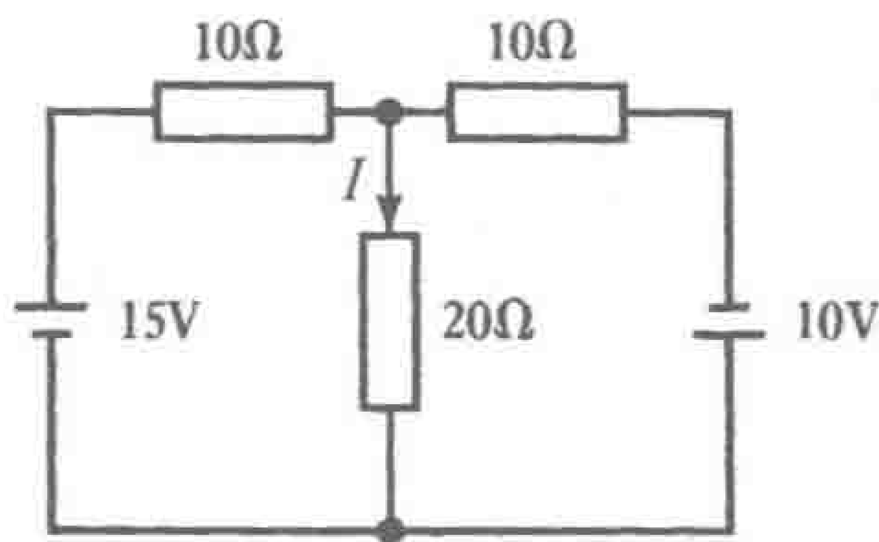


图 G-8　应用诺顿定理分析的电路

重新画的电路如图 G-9a 所示，网络 A 和 B 相连，且网络 B 为 20Ω 的电阻，需要求它上面的电流。然后求网络 A 的诺顿等效电路（见图 G-9b）。将网络 A 的两端短路得到图 G-9c 所示的电路。考虑电流的方向，短路电流为包含电压源的两个支路中的电流，即 $I_{sc}=I_1-I_2$。电流 $I_1=15/10=1.5\text{A}$，由于另一支路短路，所以 $I_2=10/10=1.0\text{A}$，因此 $I_{sc}=0.5\text{A}$。当所有独立源为零时，诺顿阻抗可以通过两个端点的电阻求得（见图 G-9d）：

$$R_N=\frac{10\times10}{10+10}=5\Omega$$

诺顿等效电路如图 G-9e 所示。将其与网络 B 相连（见图 G-9f），可得到电流 I。通过电阻的压降为 $0.5\times R_{total}$，所以电流 I 为此压降除以 20，因此，

$$I=0.5\times\frac{5}{5+20}=0.1\text{A}$$

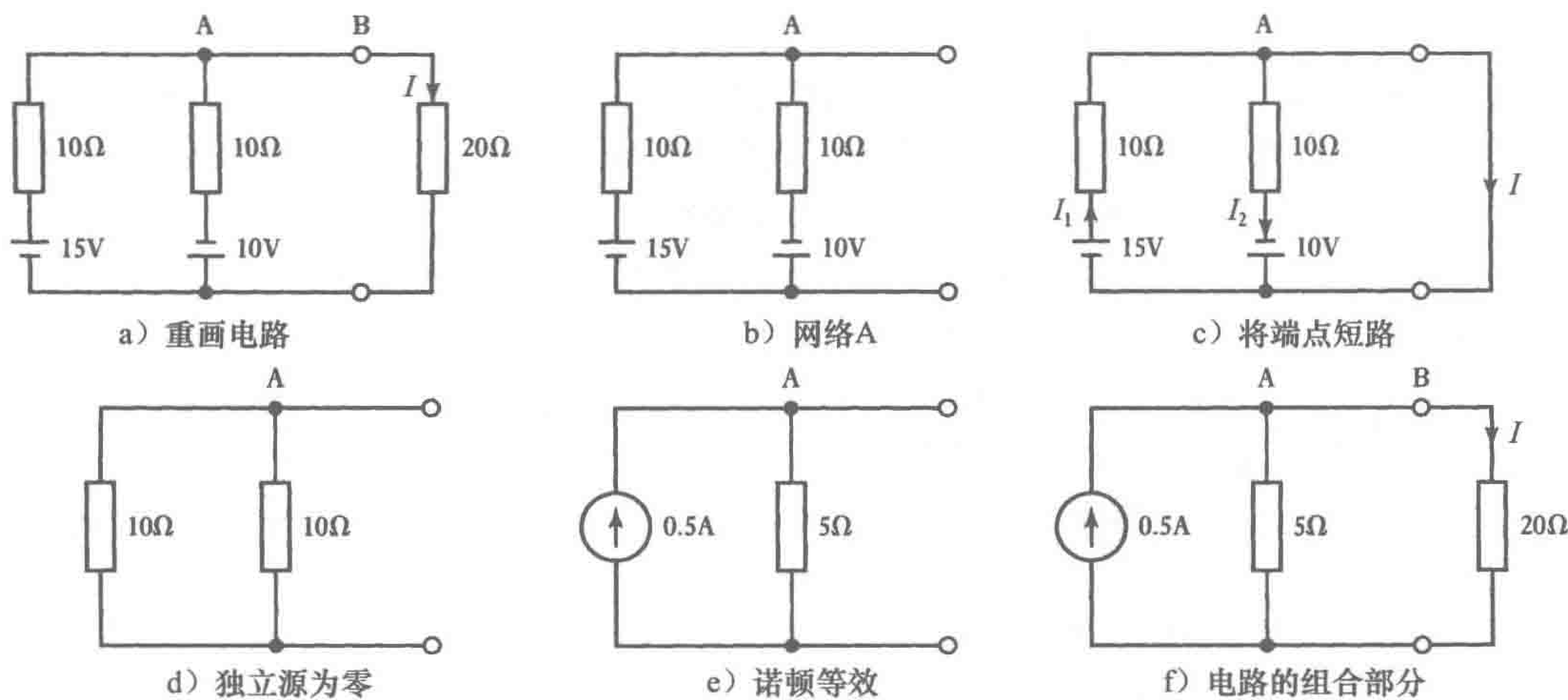


图 G-9 诺顿分析

G.2 交流电路

614 ~ 615

以恒定的角速度 ω 旋转一个半径为 OA 的线段，可以生成一个正弦波（见图 G-10a），该线段的垂直投影 AB 随着时间以正弦规律变化。在时刻 t，线段 AB 的角度 θ 为 ωt，旋转的频率 f 为 $1/T$，且 T 为旋转一周的时间，故 $\omega = 2\pi f$。在图 G-10a 中，旋转线段 OA 在 $t = 0$ 时从水平位置开始运动。图 G-10b 表明在 $t = 0$ 时 OA 已经有了一定的角度 ϕ。由于 OA 以角速度 ω 旋转，因此在时间 t 内扫过的角度为 ωt，在时刻 t 与水平的夹角为 $\omega t + \phi$。正弦变化的电流和电压可以通过这样的旋转线来描述。在初始时刻 $t = 0$，当电压、电流值为 0 时，电流为 $i = I_m \sin \omega t$，电压为 $v = V_m \sin \omega t$；当有一个初始角度 ϕ 时，电流为 $i = I_m \sin(\omega t + \phi)$，电压为 $v = V_m \sin(\omega t + \phi)$。小写表示随时间变化的电流值或电压，大写表示保持不变的值。

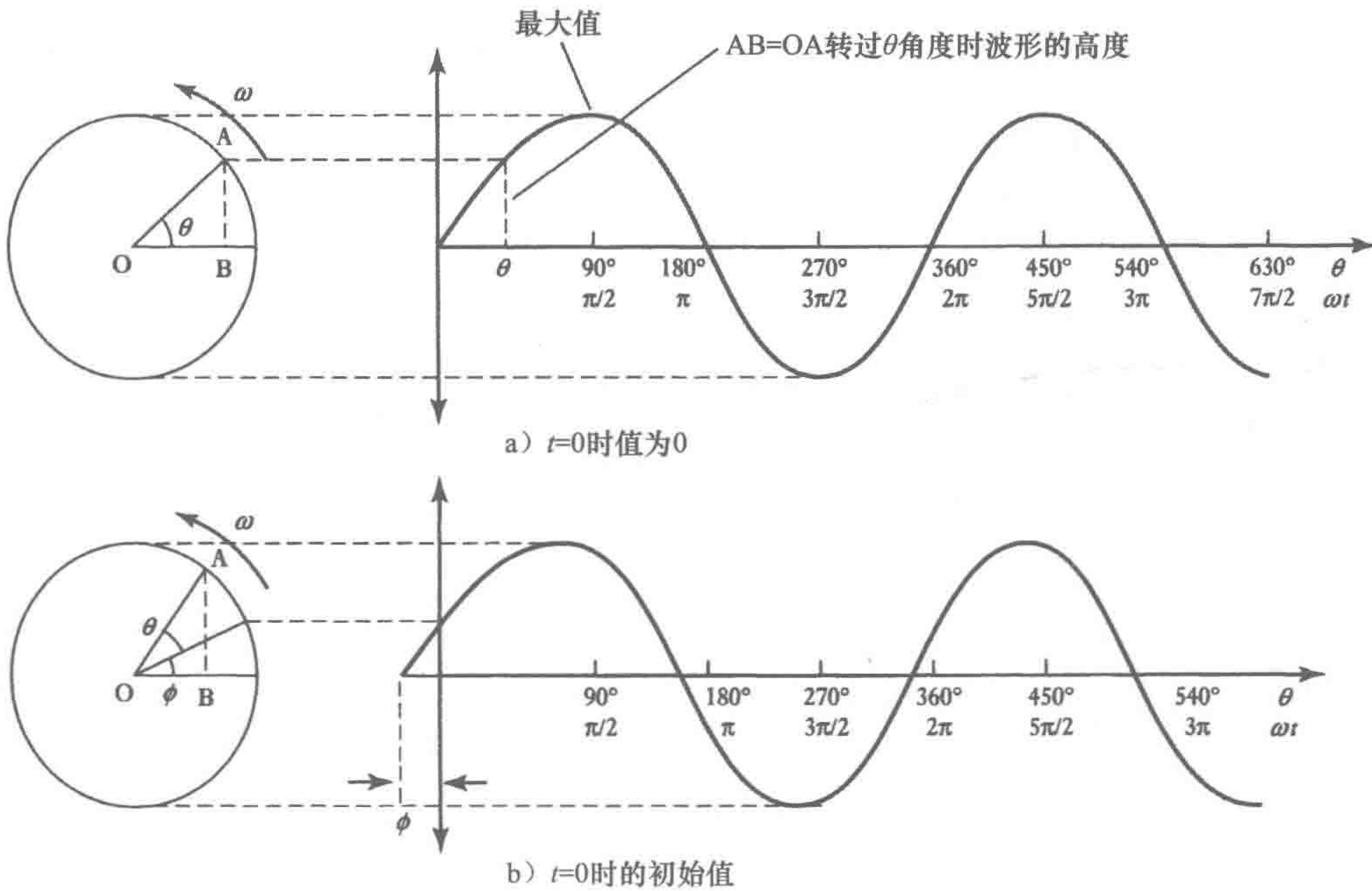


图 G-10 产生正弦波

对于交流电路，需要考虑通过某一元件的交变电流与交变电压的关系。如果将串联电路中交变电流作为参考，并用 $i = I_m \sin \omega t$ 表示，那么电压即为 $v = V_m \sin(\omega t + \phi)$。也就是说，电压和电流间有一个**相位差** ϕ。如果 ϕ 为正值，说明电压超前电流（图 G-10 中假设（a）代表电流，（b）代表电压）；如果 ϕ 为负值，则说电压**滞后**电流。

通过确定旋转线段的长度以及相对于水平参考线的初始角度，可描述一个正弦交变电流。

616

相量（phasor）是相位向量的缩写，就是用来描述这样的直线段。相量的长度代表正弦波的最大值或者方均根值（root-mean-square，r. m. s），最大值与方均根值成比例。因为同一电路中的电流和电压有相同的频率，所以表示它们的相量以相同的角速度旋转，并且时刻保持着不变的相位角。所以不用画出它们旋转的影响，画出的相量图只要给出相量间相对的角位置，忽略它们的旋转。

关于相量，其要点总结如下：

1）因为最大值与方均根值成比例，所以一个相量的长度直接与正弦交变量的最大值或者方均根值成比例。

2）相量一般以逆时针方向旋转，并且在旋转的末端有箭头。

3）两个相量间的角表示它们对应波形间的相位角。一个较大的逆时针角表明相量是超前的，较小的逆时针角表明相量是滞后的。

4）水平线是参考轴，并且相量中的一个与其同向，其他相量的角通过该参考轴得到。

G. 2. 1　交流电路中的电阻、电感和电容

考虑通过**纯电阻**的正弦电流 $i = I_m \sin \omega t$。纯电阻是指仅有电阻，没有电感或电容的元件。假设欧姆定律可以使用，通过电阻的电压为 $v = Ri$，因此 $v = RI_m \sin \omega t$。电流与电压同相，当 $\sin \omega t = 1$ 时，有最大电压 $V_m = RI_m$。

考虑通过**纯电感**的正弦电流 $i = I_m \sin \omega t$。纯电感是指只有电感而没有电阻和电容的元件。当有电感存在时，变化的电流产生反电动势 $L\mathrm{d}i/\mathrm{d}t$，其中 L 为电感。施加的电动势必须克服这一反电动势以保证电流的流动，因此通过电感的电压 v 即为 $L\mathrm{d}i/\mathrm{d}t$，即

$$v = L\frac{\mathrm{d}i}{\mathrm{d}t} = L\frac{\mathrm{d}}{\mathrm{d}t}(I_m \sin \omega t) = \omega L I_m \cos \omega t$$

因为 $\cos \omega t = \sin(\omega t + 90°)$，所以电流与电压不同相且电压超前电流 90°。当 $\cos \omega t = 1$ 时，电压取得最大值，为 $V_m = \omega L I_m$。V_m/I_m 称为**感抗** X_L，因此 $X_L = V_m/I_m = \omega L$。由于 $\omega = 2\pi f$，所以 $X_L = 2\pi f L$。可以看出感抗与频率 f 成正比，频率越高，对电流的抵抗越强。

假设施加在纯电容上的正弦电压为 $v = V_m \sin \omega t$。**纯电容**是指只有电容而没有电阻和电感的元件。电容元件上的电荷 q 与电压 v 相关，即 $q = Cv$。由于电流为电荷的移动速率 $\mathrm{d}q/\mathrm{d}t$，

617

所以有 $i = q$ 的变化速率 $= Cv$ 的变化速率 $= C \times$（v 的变化速率），即 $i = C\frac{\mathrm{d}v}{\mathrm{d}t}$。因此

$$i = \frac{\mathrm{d}q}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}(Cv) = C\frac{\mathrm{d}}{\mathrm{d}t}(V_m \sin \omega t) = \omega C V_m \cos \omega t$$

由于 $\cos \omega t = \sin(\omega t + 90°)$，所以电流与电压不同相且电压滞后电流 90°。当 $\cos \omega t = 1$ 时，电流取得最大值，为 $I_m = \omega C V_m$。V_m/I_m 称为**容抗** X_C，因此 $X_C = V_m/I_m = 1/\omega C$。容抗单位是欧姆（Ω），是阻碍电流流动的度量。容抗越大，电压要越大以使电流通过。由于 $\omega = 2\pi f$，容抗与 f 成反比，所以频率越高，对电流的阻碍越小。当使用直流电时，即频率为 0，容抗无穷大且无电流。

概括来说，图 G-11 显示了对于（a）纯电阻；（b）纯电感；（c）纯电容的电压和电流

相量。

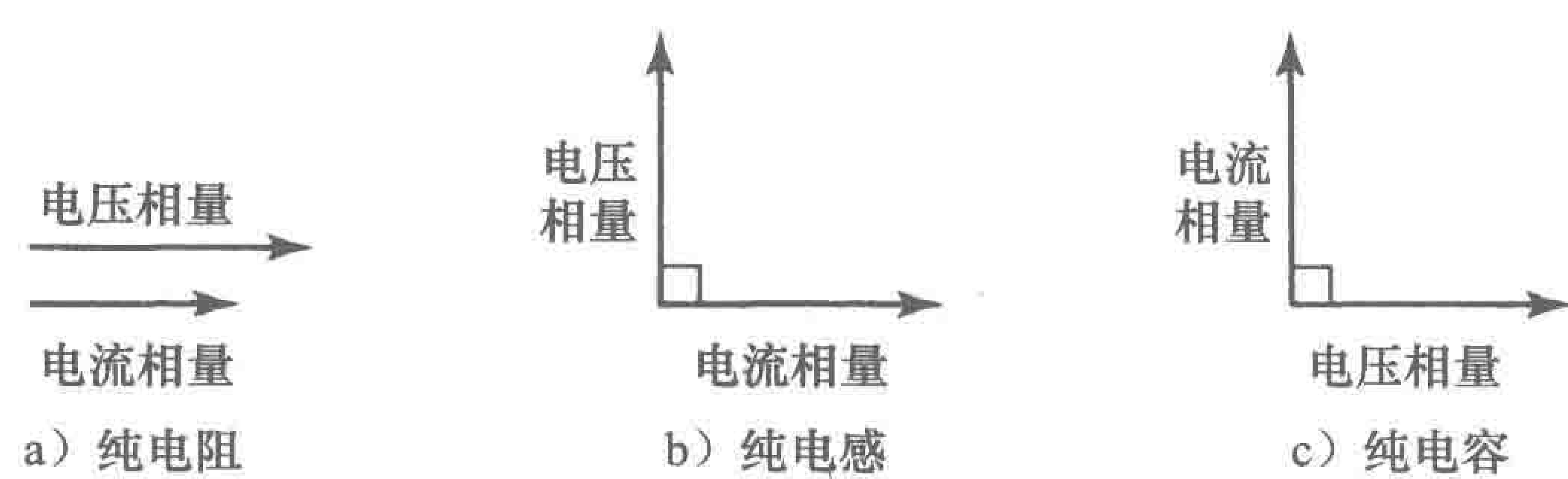


图 G-11 相量

G.2.2 交流电路的串联

对于串联电路，总的电压值是通过串联元件的电压总和，即使电压可能是不同相的。这就意味着，如果考虑相量，相量以相同的角速度旋转但可能长度不同，并以它们之间所夹的相角开始。通过相量的**平行四边形法则**（parallelogram law）可以求得两个串联电压的和：

> 如果平行四边形的邻边代表两个相量的长度和方向，那么平行四边形的对角线就是两相量的和（见图 G-12）。

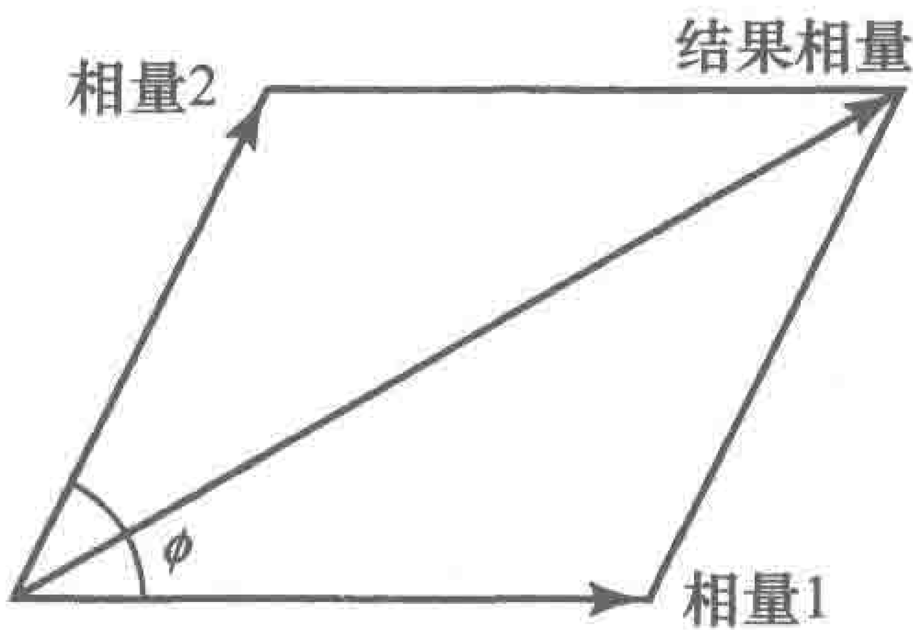


图 G-12 相差 ϕ 相位角的相量 1 和相量 2 的和

如果两个相量 V_1 和 V_2 的相位角为 90°，则由毕达哥拉斯定理（Pythagoras theorem，也称为勾股定理）可以求得 $V^2 = V_1^2 + V_2^2$，且相对于相量 V_1 的相位角 ϕ 满足 $\tan\phi = V_1/V_2$。

618

以一个电阻与电感串联的交流电路（见图 G-13a）为例来说明以上内容。对于这样的一个电路，电阻的电压和电流同向，电感的电压超前电流 90°，因此通过两个串联元件的电压之和在图 G-13b 中给出，电压相量有一个相位角 ϕ。利用勾股定理给出电压的幅值 V，即 $V^2 = V_R^2 + V_L^2$，同时用三角法也可得到相位角 ϕ，即电压超前电流的角度由 $\tan\phi = V_L/V_R$ 或 $\cos\phi = V_R/V$ 求出。

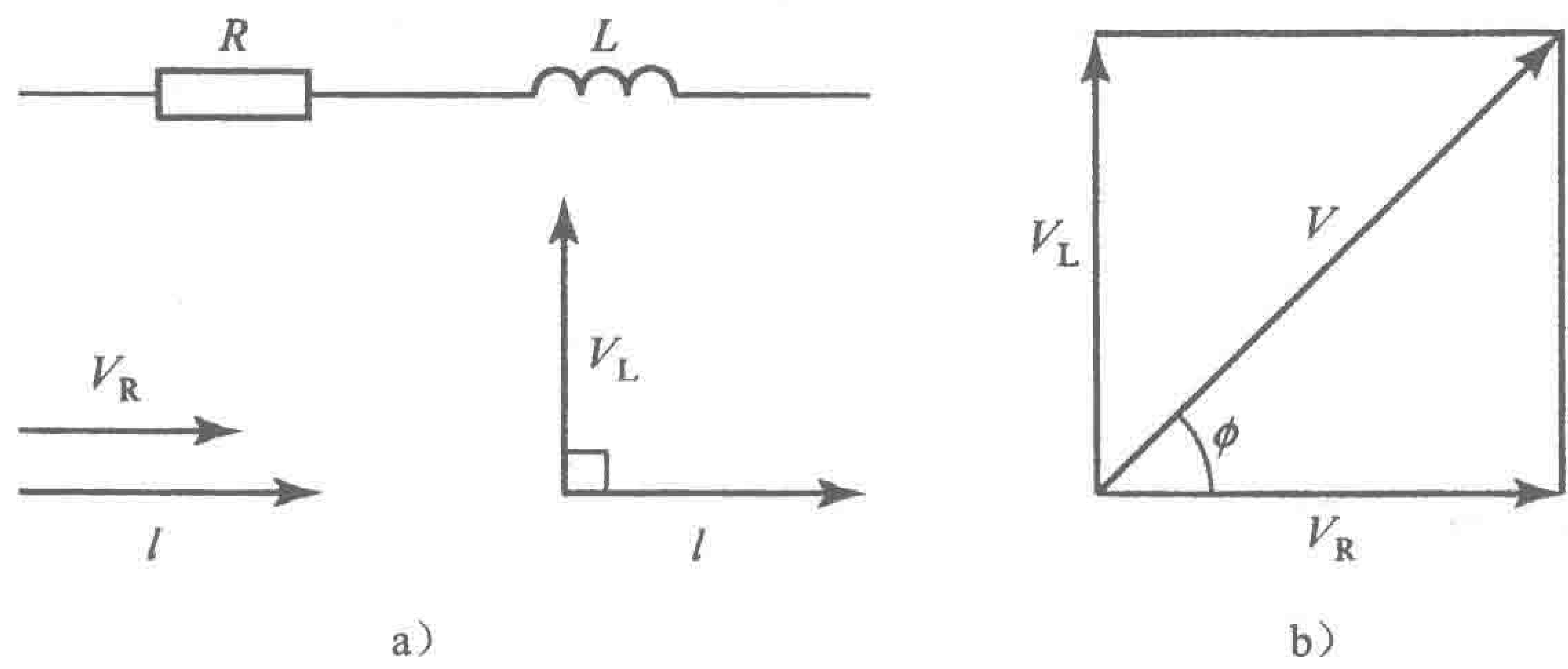


图 G-13 RL 串联电路

由于 $V_R = IR$ 且 $V_L = IX_L$，所以 $V^2 = (IR)^2 + (IX_L)^2 = I^2(R^2 + X_L^2)$。**阻抗** Z 是用来描述电路对电流的阻碍作用，并定义其为 $Z = V/I$，单位是 Ω。所以，对于串联的电阻和电感，电路的阻抗为

$$Z = \sqrt{R^2 + X_L^2} = \sqrt{R^2 + (\omega L)^2}$$

619

进一步的参考信息

下列参考书可以为学习机械电子学提供更深入的、有用的信息。

传感器和信号调理

Boyes, W., *Instrumentation Reference Book*, Newnes 2002

Clayton, G. B. and Winder, S., *Operational Amplifiers*, Newnes 2003

Figliola, R.S. and Beasley, D. E., *Theory and Design for Mechanical Measurements*, John Wiley 2000, 2005, 2011

Fraden, J., *Handbook of Modern Sensors*, Springer 2001, 2004, 2010

Gray, P. R., Hurst, P. J., Lewis, S.H. and Meyer, R. G., *Analysis and Design of Analog Integrated Circuits*, Wiley 2009

Holdsworth, B., *Digital Logic Design*, Newnes 2000

Johnson, G. W. and Jennings, R., *LabVIEW Graphical Programming*, McGraw-Hill 2006

Morris, A.S., *Measurement and Instrumentation Principles*, 3rd edition, Newnes 2001

Park, J. and Mackay, S., *Practical Data Acquisition for Instrumentation and Control Systems*, Elsevier 2003

Travis, J. and Kring, J., *LabVIEW for Everyone*, Prentice-Hall 2006

驱动

Bolton, W., *Mechanical Science*, Blackwell Scientific Publications 1993, 1998, 2006

Gottlieb, I. M., *Electric Motors and Control Techniques*, TAB Books, McGraw-Hill 1994

Kenjo, T. and Sugawara, A., *Stepping Motors and their Microprocessor Controls*, Clarenden Press 1995

Manring, N., *Hydraulic Control Systems*, Wiley 2005

Norton, R. L., *Design of Machinery*, McGraw-Hill 2003

Pinches, M. J. and Callear, B. J., *Power Pneumatics*, Prentice-Hall 1996

Wildi, T., *Electrical Machines, Drives and Power Systems*, Pearson 2005

系统模型

Åstrom, K. J. and Wittenmark, B., *Adaptive Control*, Dover 1994, 2008

Attaway, S., *Matlab: A Practical Introduction to Programming and Problem Solving*, Butterworth-Heinemann 2009

Bennett, A., *Real-time Computer Control*, Prentice-Hall 1993

Bolton, W., *Laplace and z-Transforms*, Longman 1994

Bolton, W., *Control Engineering*, Longman 1992, 1998

Bolton, W., *Control Systems*, Newnes 2002

D'Azzo, J. J., Houpis, C. H. and Sheldon, N., *Linear Control System Analysis and Design with Matlab*, CRC Press 2003

Dorf, R. C. and Bishop, H., *Modern Control Systems*, Pearson 2007

Fox, H. and Bolton, W., *Mathematics for Engineers and Technologists*, Butterworth-Heinemann 2002

Hahn, B., *Essential MATLAB for Engineers and Scientists*, 5th ed. Elsevier 2012

Moore, H., *MATLAB for Engineers*, Pearson 2013

微处理器系统

Arduino 网址：www. arduino. cc

Barnett, R. H., *The 8051 Family of Microcontrollers*, Prentice-Hall 1994
Barrett, S. F., *Arduino Microcontroller Processing for Everyone!*, Morgan & Claypool Publishers 2013
Bates, M., *PIC Microcontrollers*, Newnes 2000, 2004
Blum, J., *Exploring Arduino: Tools and Techniques for Engineering Wizardy*, Wiley 2013
Bolton, W., *Microprocessor Systems*, Longman 2000
Bolton, W., *Programmable Logic Controllers*, Newnes 1996, 2003, 2006, 2009
Cady, F. M., *Software and Hardware Engineering: Motorola M68HC11*, OUP 2000
Calcutt, D., Cowan, F. and Parchizadeh, H., *8051 Microcontrollers: An Application Based Introduction*, Newnes 2004
Ibrahim, D., *PIC Basic: Programming and Projects*, Newnes 2001
Johnsonbaugh, R. and Kalinn, M., *C for Scientists and Engineers*, Prentice Hall 1996
Lewis, R. W., *Programming Industrial Control Systems Using IEC 1131-3*, The Institution of Electrical Engineers 1998
Monk, S., *Programming Arduino*, McGraw Hill 2012
Morton, J., *PIC: Your Personal Introductory Course*, Newnes 2001, 2005
Parr, E. A., *Programmable Controllers*, Newnes 1993, 1999, 2003
Pont, M. J., *Embedded C*, Addison-Wesley 2002
Predko, M., *Programming and Customizing the PIC Microcontroller*, Tab Electronics 2007
Rohner, P., *Automation with Programmable Logic Controllers*, Macmillan 1996
Spasov, P., *Microcontroller Technology: The 68HC11*, Prentice-Hall 1992, 1996, 2001
Vahid, F. and Givargis, T., *Embedded System Design*, Wiley 2002
Van Sickle, T., *Programming Microcontrollers in C*, Newnes 2001
Yeralan, S. and Ahluwalia A., *Programming and Interfacing the 8051 Microcontroller*, Addison-Wesley 1995
Zurrell, K., *C Programming for Embedded Systems*, Kindle Edition 2000

电子系统

Storey, N., *Electronics A Systems Approach* 5th Edition Pearson 2013

这本教材包括两部分：电路和元件，电子系统。

621

专门编写了与教材配套的学生资源，这些资源可通过网址 www. pearsoned. co. uk/storey-elec 访问，点击相应标题可访问视频教程，其中包括一些机械电子学相关的内容。

3A: Kirchhoff's laws
3B: Nodal analysis
3C: Mesh analysis
3D: Selecting circuit analysis techniques
4A: Capacitors in series and in parallel
5A: Transformers
5B: Applications of inductive sensors
6A: Alternating voltages and currents
6B: Using complex impedance
7A: Power in circuits with resistance and reactance
7B: Power factor correction for an electric motor
8A: Bode diagrams
9A: Initial and final value theorems
9B: Determination of a circuit's time constant
10A: Selection of a motor for a given application
11A: Top-down system design
11B: Identification of system inputs and outputs
12A: Optical position sensors
12B: Selecting a sensor for a given application
13A: A comparison of display techniques
13B: Selecting an actuator for a given application

14A: Modelling the characteristics of an amplifier
14B: Power gain
14C: Differential amplifiers
14D: Specifying an audio amplifier
15A: Feedback systems
15B: Negative feedback
15C: Open-loop and closed-loop systems
16A: Basic op-amp circuits
16B: Some additional useful op-amp circuits
16C: Frequency response of op-amp circuits
16D: Input and output resistance of op-amp circuits
16E: Analysis of op-amp circuits
18A: Simple FET amplifiers
18B: Small signal equivalent circuit of an FET amplifier
18C: A negative feedback amplifier based on a DE MOSFET
18D: A switchable gain amplifier
19A: A simple bipolar transistor amplifier
19B: Analysis of a simple bipolar transistor amplifier
19C: Analysis of a feedback amplifier based on a bipolar transistor
19D: A common collector amplifier
19E: Design of a phase splitter
20A: Push-pull amplifiers
20B: Amplifier classes
20C: Power amplifiers
20D: Design of an electrically operated switch
21A: Simplified circuit of a bipolar operational amplifier
622
21B: Simplified circuit of a CMOS operational amplifier
21C: Circuit of a 741 operational amplifier
22A: Noise in electronic systems
24A: Binary quantities and variables
24B: Logic gates
24C: Combinational logic
24D: Boolean algebraic simplification
24E: Karnaugh maps
24F: Binary arithmetic
24G: Numeric and alphabetic codes
24H: A binary voting arrangement for a fault-tolerant system
25A: Sequential logic
25B: Shift registers
25C: Simple counters
25D: A digital stop watch
26A: Transistors as switches
26B: Logic families
26C: Transistor–transistor logic (TTL)
26D: Complementary metal oxide semiconductor logic (CMOS)
26E: Implementing complex gates in CMOS
26F: Interfacing logic gates of different families
26G: Power dissipation in digital systems
26H: Tackling noise-related problems in industrial systems
27A: Implementing digital systems
27B: Programmable logic array (PLA)
27C: Programmable array logic (PAL)
27D: Complex array logic
27E: Microcomputer systems
27F: Input/output techniques

27G: Interrupt-driven input/output
27H: Motor control in a washing machine
28A: Sampling
28B: Digital-to-analogue conversion
28C: Multiplexing
28D: Data acquisition and conversion in a smartphone
29A: Communications systems
29B: Analogue modulation
29C: Digital and pulse modulation
30B: A control system for a robotic arm

623

习题参考答案

下面提供的是对数值习题的答案和对描述性习题的简单提示。

第1章

1.1 （a）传感器，汞；信号调节器，细口杆；显示，杆上的标志

（b）传感器，弧形管；信号调节器，齿轮；显示，沿刻度盘移动的指针

1.2 见正文

1.3 比较单元/控制器，恒温器；校正单元，继电器；过程，加热；变量，温度；测量，温度敏感装置或双金属片

1.4 见图 P-1

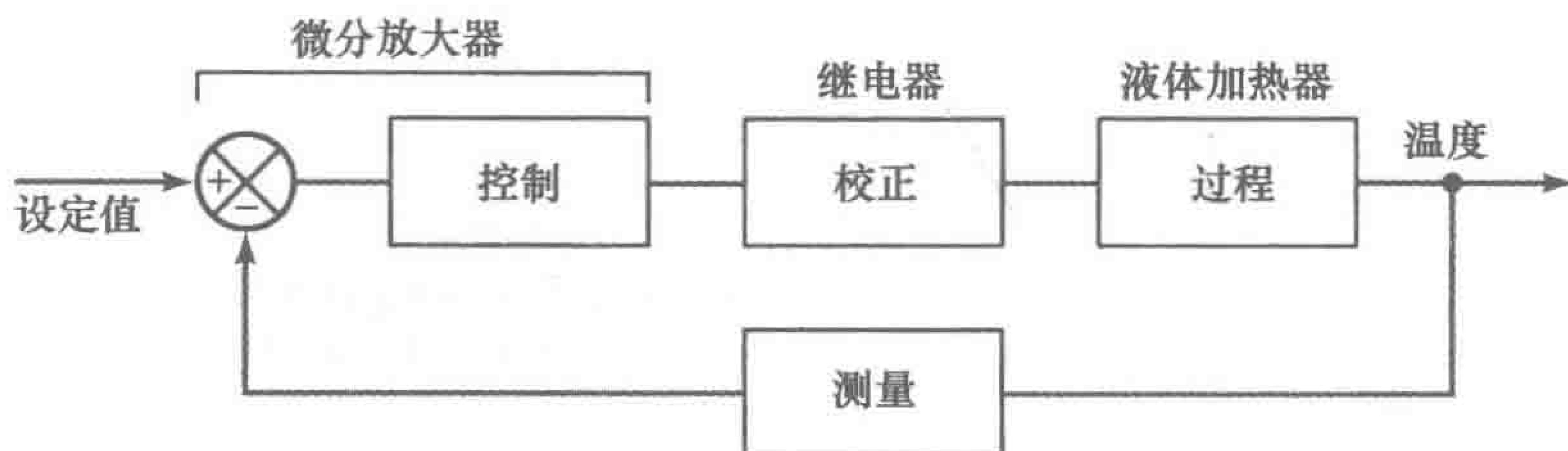


图 P-1　习题 1.4

1.5 见正文

1.6 见正文

1.7 例如：进水，漂洗，出水，进水，加热水，漂洗，出水，进水，漂洗，出水

1.8 传统的：体积大，功能有限，需要重复上弦。机械电子的：结构紧凑，功能多，无须重复上弦，便宜

1.9 双金属元件：慢，精度有限，功能单一，便宜

微处理器：快，精度高，功能多，更便宜

第2章

2.1 见书中的术语

2.2 −3.9%

2.3 76.4s

2.4 0.73%

2.5 0.105Ω

2.6 增量式：输出一系列脉冲，可相对于一个基准进行计数，不是绝对位置

绝对式：输出信号和实际角度一一对应

2.7 162

2.8 （a）±1.2°；（b）3.3mV

2.9 见正文

2.10 2.8kPa

2.11　19.6kPa

2.12　-0.89%

2.13　+1.54℃

2.14　是

2.15　-9.81N，-19.62N，例如应变计

2.16　例如，带差压传感器的孔板

2.17　例如，差压传感器

2.18　例如，LVDT 位移传感器

第3章

3.1　见图 3-2，$R_2/R_1=50$，如 $R_1=1\text{k}\Omega$，$R_2=50\text{k}\Omega$

3.2　200kΩ

3.3　图 3-5 有两个输入，如 $V_A=1\text{V}$，$V_B=0\sim100\text{mV}$，$R_A=R_2=40\text{k}\Omega$，$R_B=1\text{k}\Omega$

3.4　图 3-11，$R_1=1\text{k}\Omega$，$R_2=2.32\text{k}\Omega$

3.5　$V=K\sqrt{I}$

3.6　100kΩ

3.7　80dB

3.8　熔断器在大电流时起保护作用，限流电阻用以减小电流，二极管用于交流整流，Zener 二极管电路起电压和极性保护作用，低通滤波器可以除去噪声干扰，光隔离器把微处理器与高电压隔离

3.9　0.234V

3.10　2.1×10^{-4}V

3.11　如问题中给出

第4章

4.1　24.4mV

4.2　9

4.3　0.625V

4.4　1，2，4，8

4.5　12μs

4.6　见正文

4.7　缓冲器，数模转换器，保护

4.8　0.33V，0.67V，1.33V，2.67V

4.9　32 768R

4.10　15.35ms

4.11　请参考相关 12bitADC 的数据手册

625

第5章

5.1　例如：(a) 票的选择与正确付费 AND，正确的金额由各种可能的输入通过 OR 门来决定 (b) 和安全保护相 AND 的有润滑剂、冷却、工件、电源等的状态，包括所有的操作 (c) 图 P-2，(d) AND

5.2　(a) Q，(b) P

5.3　AND 门

5.4　A 是 1，B 是 0

5.5 见图 P-3

5.6 见图 P-4

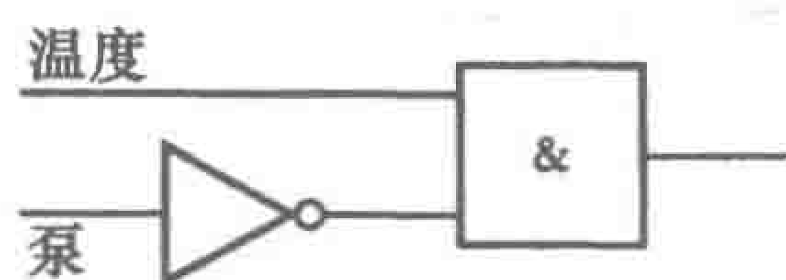


图 P-2 习题 5.1(c)

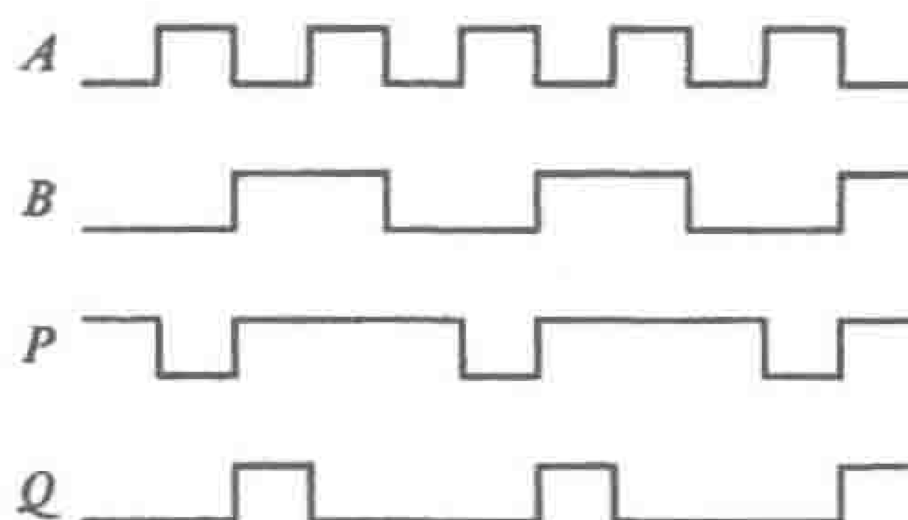


图 P-3 习题 5.5

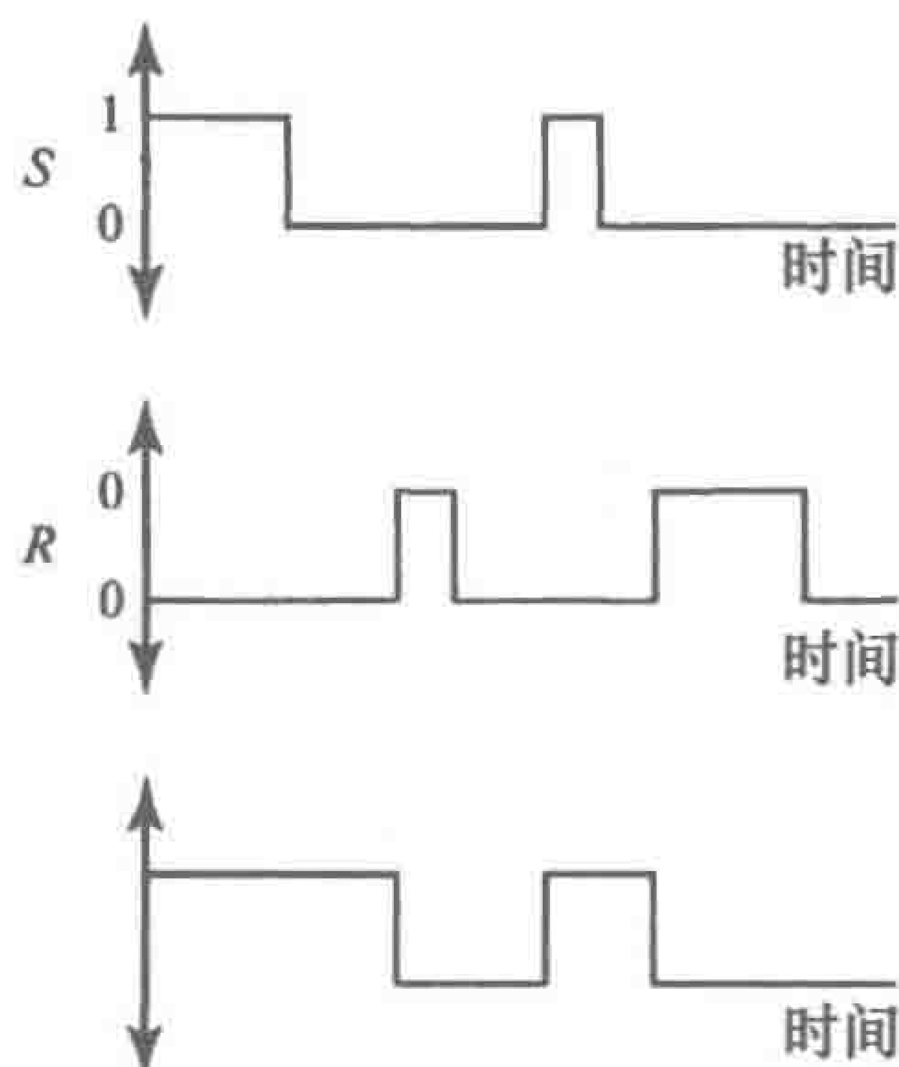


图 P-4 习题 5.6

5.7 见正文 5.3.1 节，交叉耦合 NOR 门

第 6 章

6.1 见正文

6.2 见 6.1 节

6.3 例如：(a) 记录仪，(b) 动圈式仪表，(c) 硬盘或 CD，(d) 存储示波器或硬盘或 CD

6.4 可以是四主动臂桥，微分运算放大器，电压表显示。元件的值由所选钢的厚度和称重传感器的直径决定。可以选择把水箱安装在三个称重传感器上。

6.5 如图 3-8 所示，可以用电桥作为冷端补偿（见 3.5.2 节），通过合适的热电偶材料的选择来实现线性化。

6.6 可采用带采样保持元件的热敏电阻，之后每个传感器连接模数转换器。这将得到用于传输的数字信号，以便减少可能存在的干扰。光隔离器可以用于隔离的高电压/电流，然后由多路开关送给数字仪表。

6.7 基于阿基米德原理：向上的浮力等于排开液体的重力。

6.8 可以用 LVDT 或应变计并采用惠斯通电桥。

6.9 例如：(a) 波登管仪表，(b) 热敏电阻，电流计图表记录仪，(c) 应变计重力传感器，惠斯通电桥，差动放大器，数字电压表，(d) 测速发电动机，对脉冲形状的信号调理，计数器

626

第 7 章

7.1 见 7.3 节

7.2 见 7.3.2 节

7.3 见 7.4 节

7.4 见 (a) 图 7-14，(b) 图 7-8b，(c) 图 7-10，(d) 图 7-13

7.5 A+，B+，A-，B-

7.6 见图 P-5

7.7 $0.0057m^2$

7.8 124mm

7.9 1.27MPa，$3.9 \times 10^{-5} m^3/s$

7.10 (a) $0.05m^3/s$，(b) $0.10m^3/s$

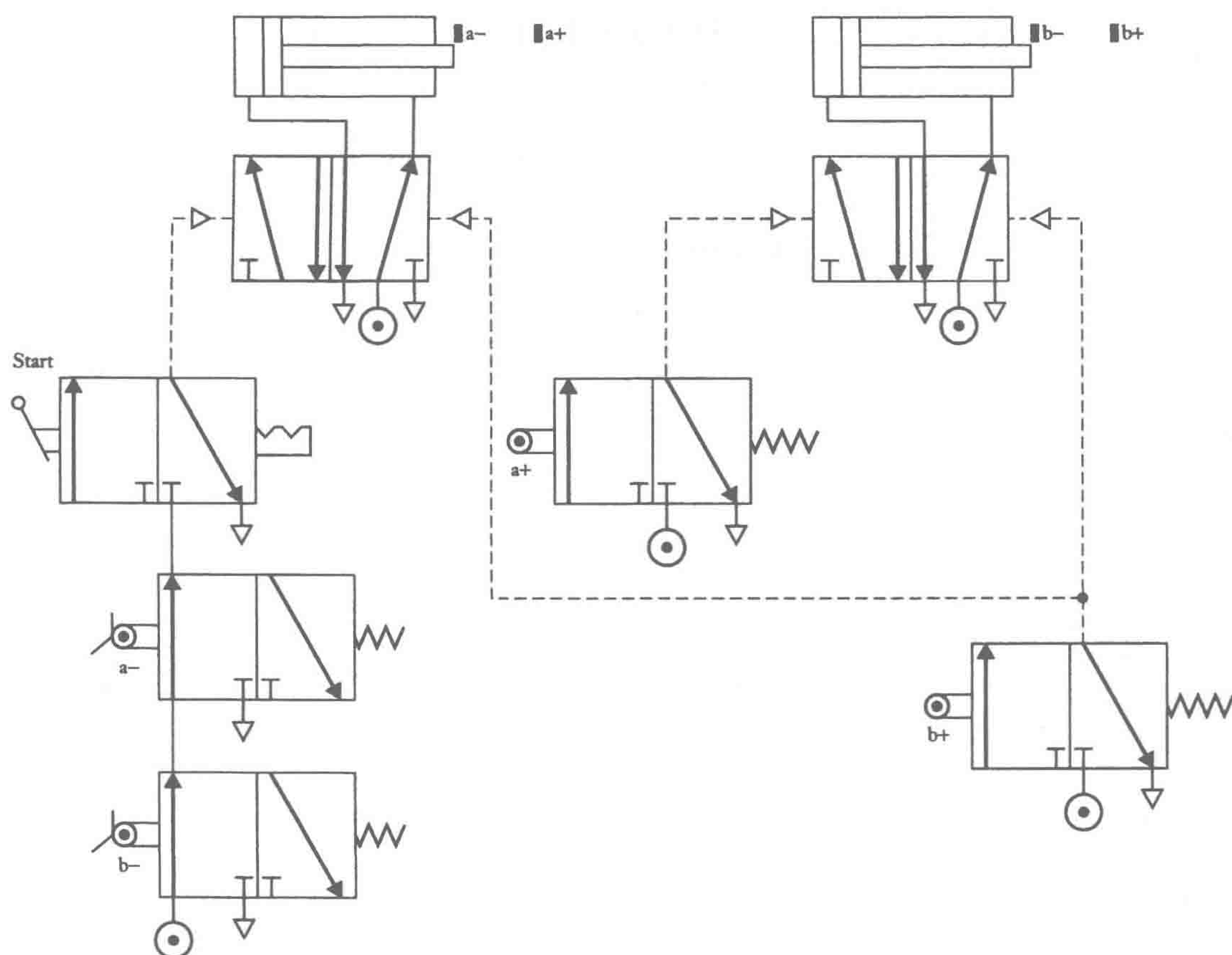


图 P-5　习题 7.6

7.11　(a) $0.42m^3/s$, (b) $0.89m^3/s$

7.12　960mm

第 8 章

8.1　(a) 用来将一种运动形式转换成另一种运动形式的系统
　　(b) 对给定的输入运动提供可控输出的一系列的铰和运动副

8.2　见 8.3.1 节

8.3　(a) 1, (b) 2, (c) 1, (d) 1, (e) 3

627

8.4　(a) 纯平动, (b) 纯平动, (c) 纯转动, (d) 纯转动, (e) 平动与转动的叠加

8.5　急回

8.6　突然产生向下的位移, 然后又渐渐地升起

8.7　60mm

8.8　见图 8-14a 的心形, 从转动轴到心形顶部和底部的距离分别为 40mm 和 100mm

8.9　例如: (a) 在一个轴上的凸轮, (b) 急回机构, (c) 偏心凸轮, (d) 齿条齿轮, (e) 带传动, (f) 锥齿轮

8.10　1/24

第 9 章

9.1　其作用类似于触发器

9.2　见正文和图 9-7

9.3　(a) 串励, (b) 并励

9.4 (a) 直流并励，(b) 有逆变器的感应或同步电动机，(c) d.c.，(d) a.c.

9.5 见9.5.4节

9.6 见9.7节

9.7 480脉冲/秒

9.8 9°

9.9 (a) 4kW，(b) 800W，(c) 31.8Nm

9.10 0.65Nm

9.11 2

9.12 3.6Nm

第10章

10.1 见10.2节

10.2 256

10.3 64K×8

10.4 见10.3节

10.5 见图10-9和相关的文字

10.6 (a) E，(b) C，(c) D，(d) B

10.7 256

10.8 (a) 0，(b) 1

10.9 见10.3.1节，第6项

10.10 见10.3.2节，第5项

10.11 复位引脚高电平

10.12 (a)

```
IF A
   THEN
     BEGIN B
     END B
ELSE
     BEGIN C
     END C
ENDIF A
```

628

(b)

```
WHILE A
  BEGIN B
    END B
ENDWHILE A
```

第11章

11.1 (a) 89，(b) 99

11.2 不必指定地址，因为助记符隐含了地址

11.3 (a) CLRA，(b) STAA，(c) LDAA，(d) CBA，(e) LDX

11.4 (a) LDAA $20，(b) DECA，(c) CLR $0020，(d) ADDA $0020

11.5 (a) 将累加器B中的值存到地址0035中，(b) 累加器A装载数据F2
(c) 清除进位标志，(d) 累加器A中的值加1，(e) 将累加器A中的值与C5比较
(f) 清空地址2000中的内容，(g) 跳转到由变址寄存器加05所给出的地址

11.6 (a)

```
DATA1       EQU         $0050
DATA2       EQU         $0060
```

```
DIFF    EQU     $0070
        ORG     $0010
        LDAA    DATA1       ;取得被减数
        SUBA    DATA2       ;减去减数
        STAA    DIFF        ;存储所得的差
        SWI                 ;程序结束
```

(b)

```
MULT1   EQU     $0020
MULT2   EQU     $0021
PROD    EQU     $0022
        ORG     $0010
        CLR     PROD        ;清零乘积地址
        LDAB    MULT1       ;取得第一个数
SUM     LDAA    MULT2       ;取得被乘数
        ADDA    PROD        ;被乘数加到累加器 A
        STAA    PROD        ;存储结果
        DECB                ;累加器 B 减 1
        BNE     SUM         ;如果没加完,转移
        WAI                 ;程序结束
```

(c)

```
FIRST   EQU     $0020
        ORG     $0000
        CLRA                ;清空累加器 A
        LDX     #0
MORE    STAA    $20,X
        INX                 ;变址寄存器加 1
        INCA                ;累加器加 1
        CMPA    #$10        ;与数值 10 进行比较
        BNE     MORE        ;如果不为 0,转移
        WAI                 ;程序结束
```


(d)

```
        ORG     $0100
        LDX     #$2000      ;设置指针
LOOP    LDA A   $00,X       ;加载数据
        STA A   $50,X       ;存储数据
        INX                 ;变址寄存器加 1
        CPX     $3000       ;比较
        BNE     LOOP        ;转移
        SWI                 ;程序结束
```

11.7

```
YY      EQU     $??         ;选择需要的延时时间
SAVEX   EQU     $0100
        ORG     $0010
        STA     SAVEX       ;保存累加器 A
        LDAA    YY          ;加载累加器 A
LOOP    DECA                ;A - $01
        BNE     LOOP        ;如果不为 0,转移
        LDA     SAVEX       ;恢复累加器
        RTS                 ;返回调用程序
```

11.8

```
LDA       $2000     ;读输入数据
AND A     #$01      ;除了位0外屏蔽所有位
BEQ       $03       ;如果开关为低电平,则跳过占三个程序行的JMP指令
JMP       $3000     ;如果开关为高电平,则不跳转,执行JMP
Continue
```

第12章

12.1 (a) 变量counter定义成integer（整型数），(b) 变量num赋值为10，(c) Name将显示，(d) 显示Number 12，(e) 包含文件stdio.h

12.2 (a) 调用stdio.h库中的printf() 函数，(b) 说明一组语句的开头和结尾，(c) 另起一行，(d) 问题3

12.3 数是12

12.4
```
#include <stdio.h>
int main(void);
{
    Int len,width;
    printf("Enter Length:");
    scanf("%d",&len);
    printf("Enter width:");
    scanf("%d",&width);
    printf("Area is %d",lens * width);
    return 0;
}
```

12.5 类似于12.3节中的第4项

630 12.6 除非第二个数是0，否则将第一个数除以第二个数

第13章

13.1 见13.3节

13.2 见13.3节。并行接口有和微处理器数目相同的输入/输出线。串行接口只有单一的输入/输出线

13.3 见13.2节

13.4 见13.4节

13.5 见13.4节和图13-10

13.6 见13.4.1节

13.7 见13.3.3节。循环扫描方式是对所有外部设备以一定频率间隔进行访问，包括那些没有使用的外设。因此这很浪费时间。中断只在外设请求时才被启用，因此具有更高的效率。

13.8 CRA 00110100，CRB 00101111

13.9 11.4.2节中的程序把LDAA #$05转换成LDAA #$34，LDAA #$34转换成LDAA #$2F

13.10 如13.4.2节中的程序后接：

```
READ LDAA $2000; READ port A
```

可能一定的延时之后为：

```
BRA  READ
```

第14章

14.1 (a) AND，(b) OR

14.2 (a) 图14-9b，(b) 图14-10b，(c) 一个锁存电路，如图14-16，输入1是启动开关，输入2是停止开关

631

14.3　0 LD X400，1 LD Y430，2 ORB，3 ANI X401，4 OUT Y430

14.4　0 LD X400，1 OR Y430，3 OUT Y430，4 OUT T450，5 K 50；延时接通定时器

14.5　0 LD X400，1 OR Y430，2 ANI M100，3 OUT Y430，4 LD X401，5 OUT M100；复位锁存电路。

14.6　见图 14-28，定时器 1 中 K = 1 延时 1s，定时器 2 中 K = 20 延时 20s。

14.7　图 P-6

14.8　图 P-7

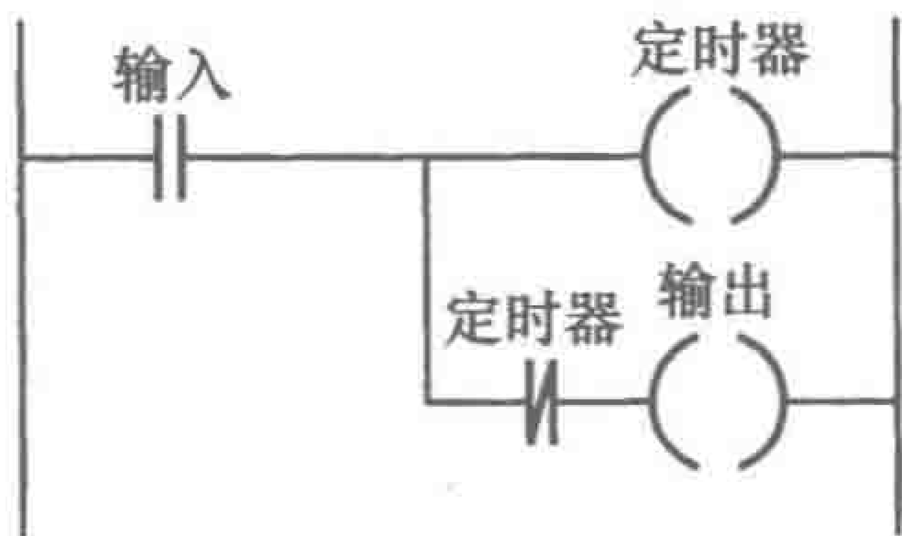


图 P-6　习题 14.7

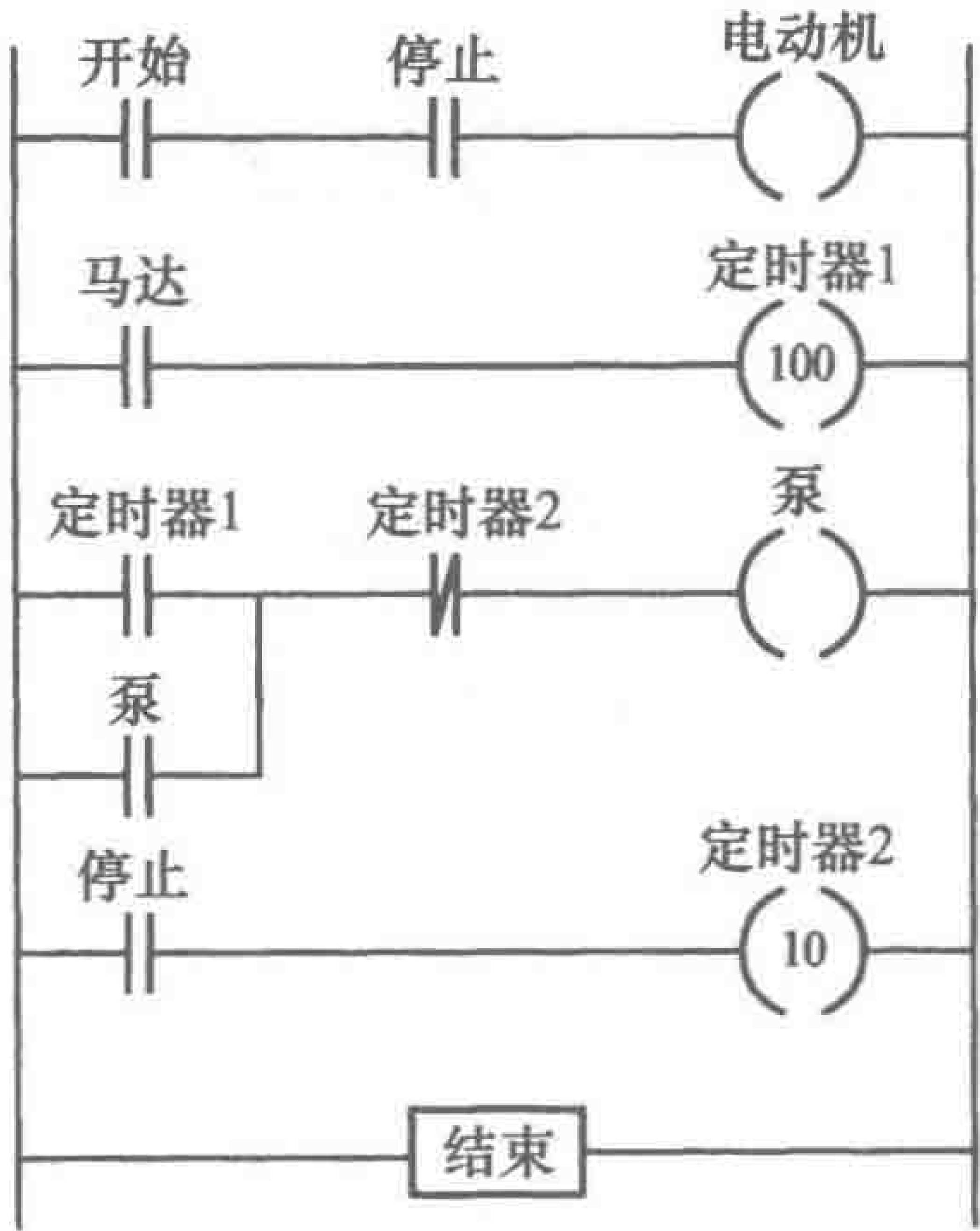


图 P-7　习题 14.8

14.9　图 P-8

14.10　图 P-9

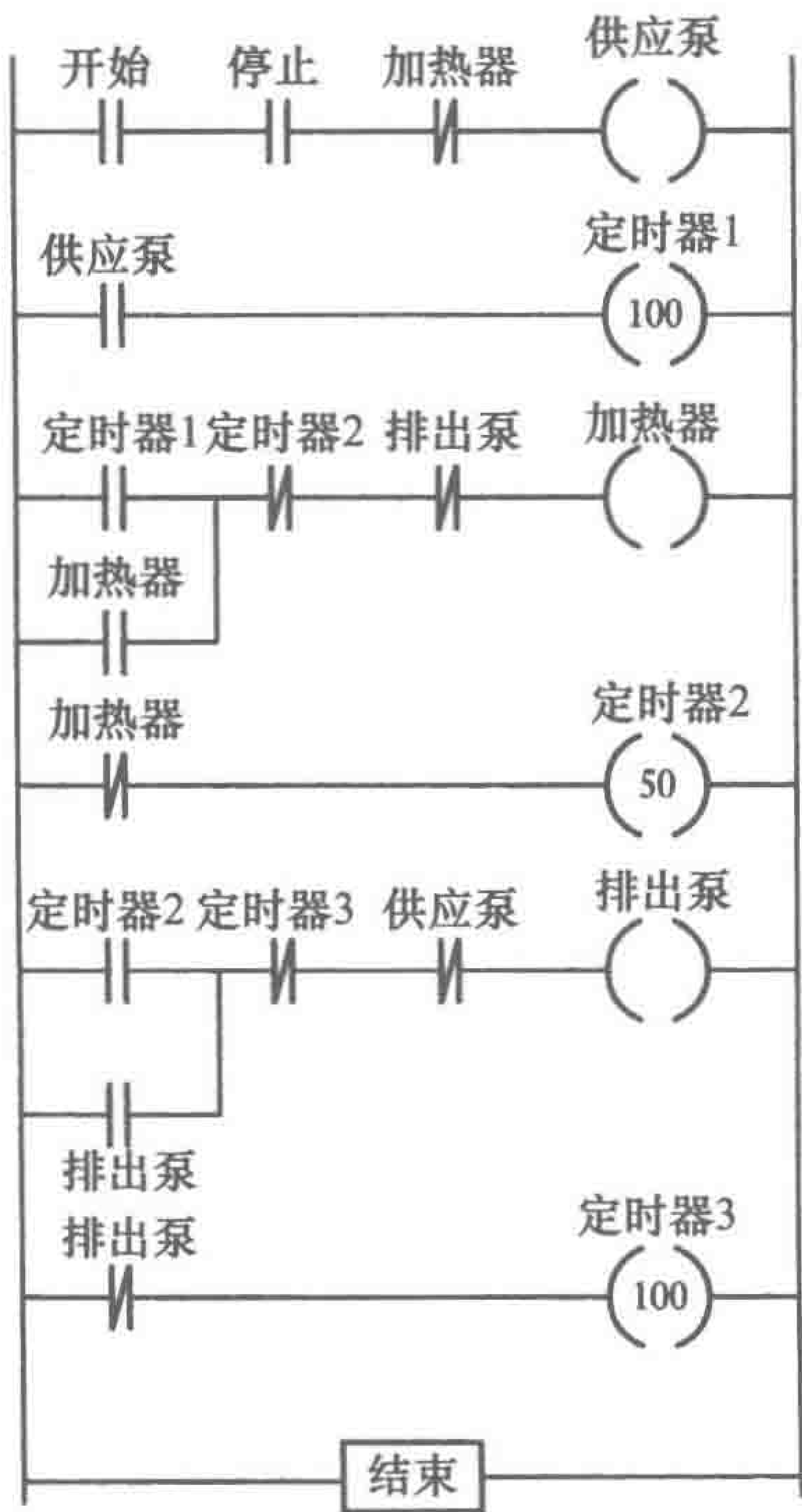


图 P-8　习题 14.9

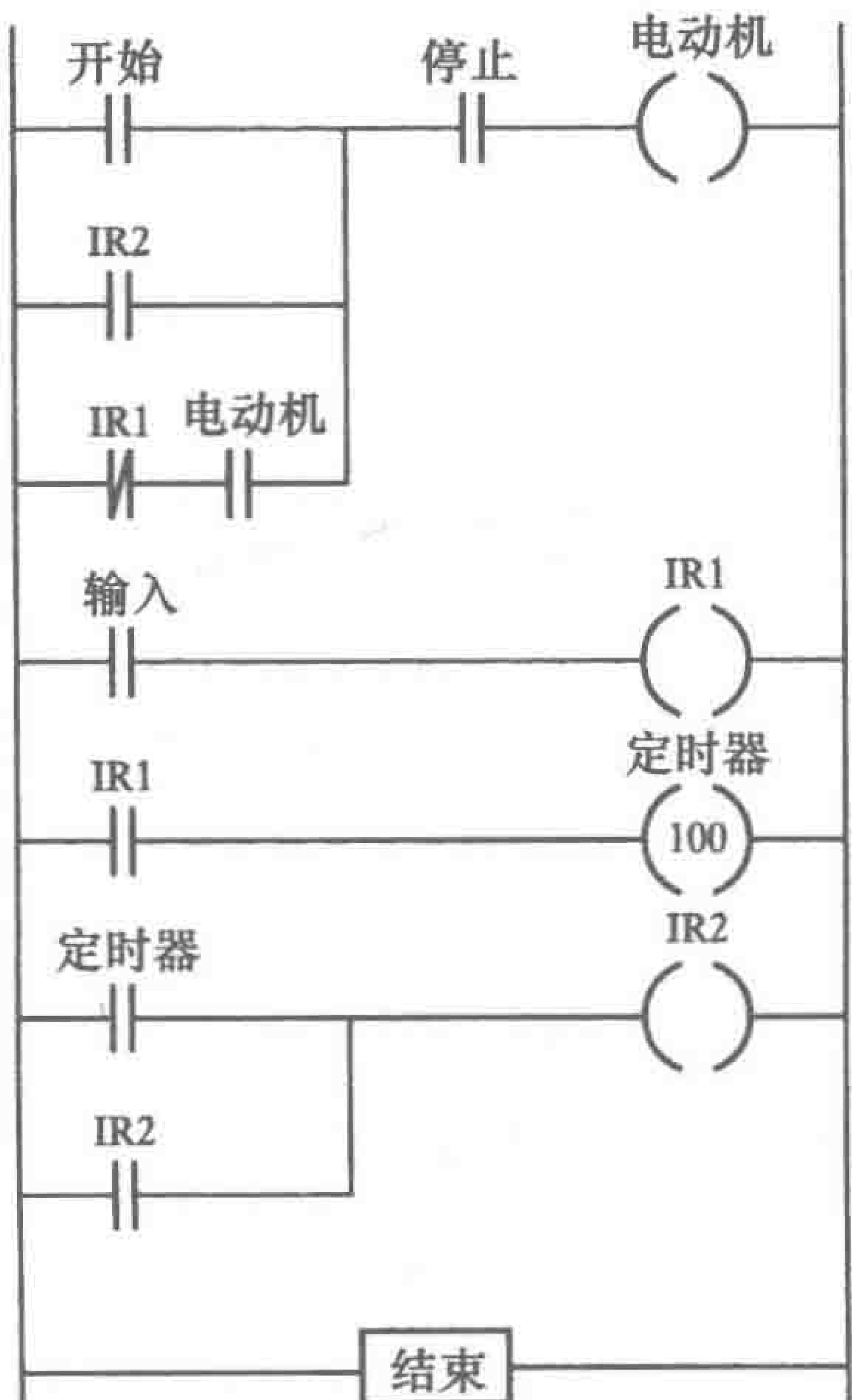


图 P-9　习题 14.10

14.11 先有输出打开，但在下一个输入到来时关闭，后一个与前一个一样。

14.12 见 14.10 节

14.13 两个锁存电路，如图 P-10 所示

14.14 图 P-11

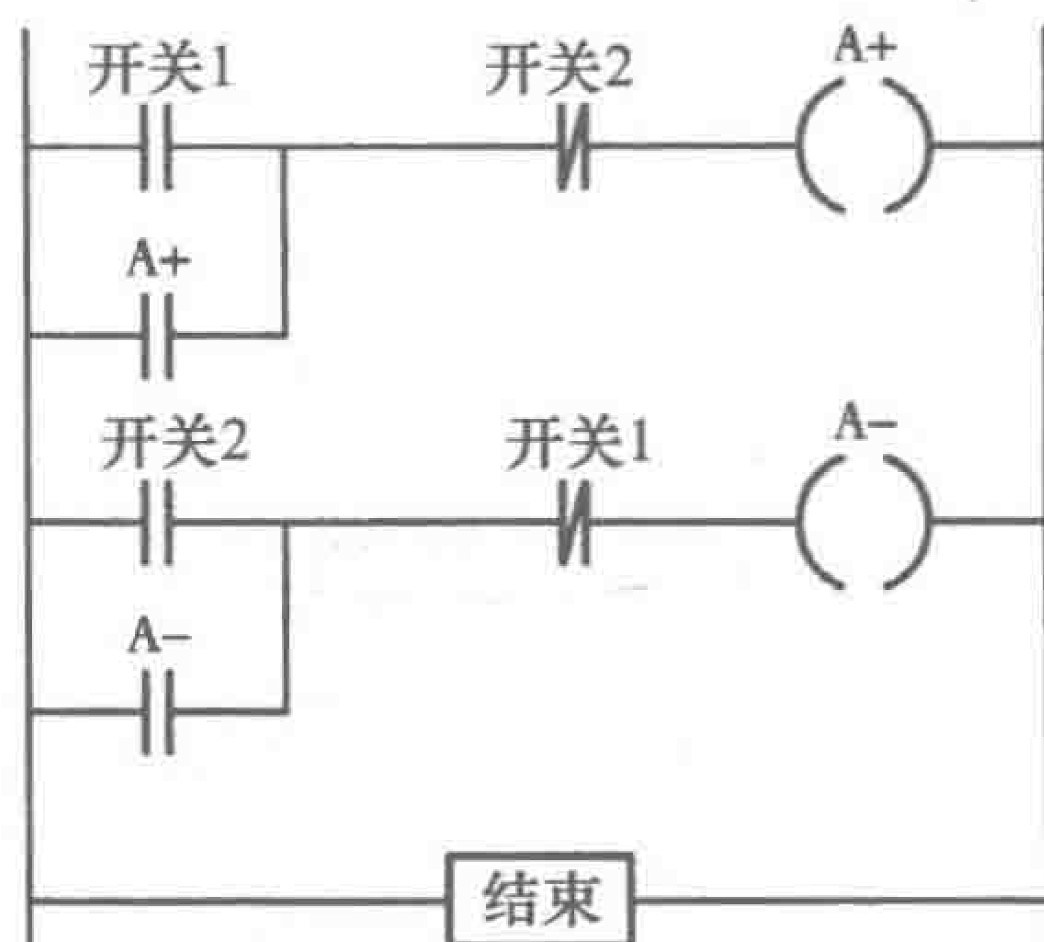


图 P-10 习题 14.13

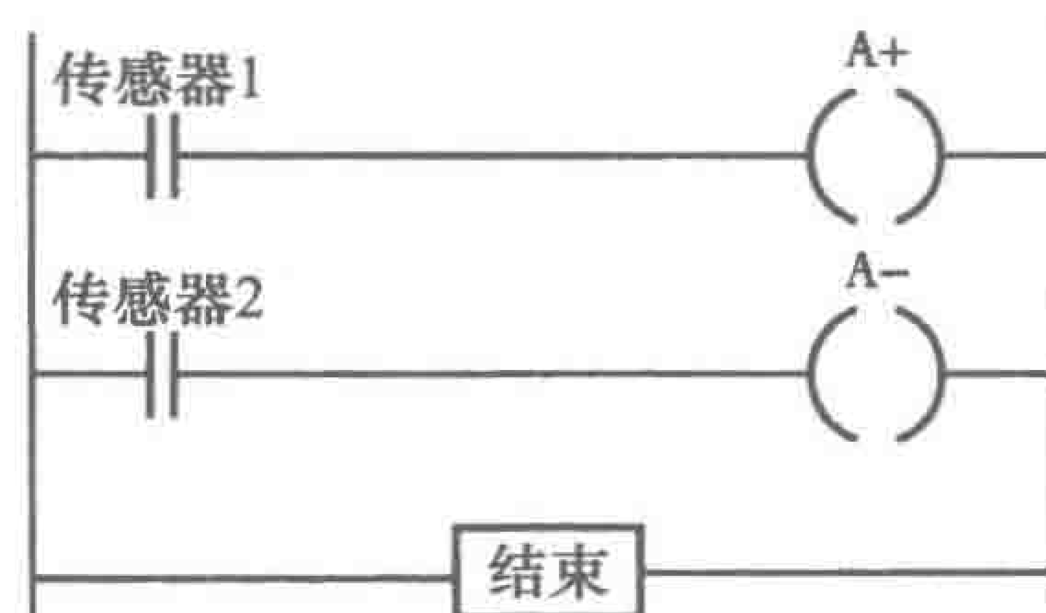


图 P-11 习题 14.14

第 15 章

15.1 见 15.2 节

15.2 见 15.3 节

15.3 总线

15.4 宽带

15.5 见 15.5.1 节

15.6 见 15.4 节

15.7 见 15.3.1 节

15.8 NRFD 连接 PD0，DAV 连接 STRA 和 IRQ，NDAC 连接 STRB，数据连接到端口 C

15.9 TTL 到 RS-232C 信号电平转换

632 15.10 见 15.7.1 节

第 16 章

16.1 见 16.1 节

16.2 见 16.2 节

16.3 见 16.2 节

16.4 见 16.5.3 节的可编程校验与校验和，以及 16.2 节的看门狗定时器。

16.5 见 16.5.3 节

第 17 章

17.1 (a) $m\frac{d^2x}{dt^2}+c\frac{dx}{dt}=F$，(b) $m\frac{d^2x}{dt^2}+c\frac{dx}{dt}+(k_1+k_2)x=F$

17.2 如图 17-3a 所示

17.3 $c\frac{d\theta_i}{dt}=c\frac{d\theta_0}{d_t}+k\theta_0$

17.4 两根扭簧与惯量块串联

$$T=I\frac{d^2\theta_0}{dt^2}+k_1(\theta_1-\theta_2)=m\frac{d^2\theta}{dt^2}+\frac{k_1k_2}{k_1+k_2}\theta_1$$

17.5 $v=v_R+\frac{1}{RC}\int v_R dt$

17.6 $v = \frac{L}{R}\frac{dv_R}{dt} + \frac{1}{CR}\int v_R dt + v_R$

17.7 $v = R_1 C \frac{dv_C}{dt} + \left(\frac{R_1}{R_2} + 1\right) v_C$

17.8 $RA_2 \frac{dh_2}{dt} + h_2 \rho g = h_1$

17.9 $RC\frac{dT}{dt} + T = T_r$。充满电的电容通过电阻放电

17.10 $RC\frac{dT_1}{dt} = Rq - 2T_1 + T_2 + T_3$，$RC\frac{dT_2}{dt} = T_1 - 2T_2 + T_3$

17.11 $pA = m\frac{d^2x}{dt^2} + R\frac{dx}{dt} + \frac{1}{C}x$，$R$=杆运动所受的阻尼，$c$=弹簧的容抗

17.12 $T = \left(\frac{I_1}{n} + n\right)\frac{d^2\theta}{dt^2} + \left(\frac{c_1}{n} + nc_2\right)\frac{d\theta}{dt} + \left(\frac{k_1}{n} + nk_2\right)\theta$

633

第18章

18.1 $\frac{IR}{k_1 k_2}\frac{d\omega}{dt} + \omega = \frac{1}{k_2}v$

18.2 $(L_a + L_L)\frac{di_a}{dt} + (R_a + R_L)i_a - k_1\frac{d\theta}{dt} = 0$，$I\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} + k_2 i_a = T$

18.3 和电枢控制电动机相同

第19章

19.1 $4\frac{dx}{dt} + x = 6y$

19.2 (a) 59.9℃，(b) 71.9℃

19.3 (a) $i = \frac{V}{R}(1 - e^{-Rt/L})$，(b) L/R，(c) V/R

19.4 (a) 连续振动，(b) 欠阻尼，(c) 临界阻尼，(d) 过阻尼

19.5 (a) 4Hz，(b) 1.25，(c) $i = I\left(\frac{1}{3}e^{-8t} - \frac{4}{3}e^{-2t} + 1\right)$

19.6 (a) 5Hz，(b) 1.0，(c) $x = (-32 + 6t)e^{-5t} + 6$

19.7 (a) 9.5%，(b) 0.020s

19.8 (a) 4Hz，(b) 0.625，(c) 1.45Hz，(d) 0.5s，(e) 8.1%，(f) 1.4s

19.9 (a) 0.59，(b) 0.87

19.10 2.4

19.11 0.09

19.12 3.93rad/s，0.63Hz

第20章

20.1 (a) $\frac{1}{As + \rho g/R}$，(b) $\frac{1}{ms^2 + cs + k}$，(c) $\frac{1}{LCs^2 + RCs + 1}$

20.2 (a) 3s，(b) 0.67s

20.3 (a) $1 + e^{-2t}$，(b) $2 + 2e^{-5t}$

20.4 (a) 过阻尼，(a) 欠阻尼，(c) 临界阻尼，(d) 欠阻尼

20.5 te^{-3t}

20.6 $2e^{-4t} - 2e^{-3t}$

634

20.7 (a) $\frac{4s}{s^2(s+1)+4}$，(b) $\frac{2(s+2)}{(s+1)(s+2)+2}$

(c) $\frac{4}{(s+2)(s+3)+20}$，(d) $\frac{2}{s(s+2)+20}$

20.8 $5/(s+53)$

20.9 $5s/(s^2+s+10)$

20.10 $2/(3s+1)$

20.11 -1，-2

20.12 (a) 稳定，(b) 不稳定，(c) 不稳定，(d) 稳定，(e) 不稳定

第21章

21.1 (a) $\frac{5}{\sqrt{\omega^2+4}}$，$\frac{\omega}{2}$，(b) $\frac{2}{\sqrt{\omega^4+\omega^2}}$，$\frac{1}{\omega}$

(c) $\frac{1}{\sqrt{4\omega^6-3\omega^4+3\omega^2+1}}$，$\frac{\omega(3-2\omega^2)}{1-3\omega^2}$

21.2 0.56 sin $(5t-38°)$

21.3 1.18 sin $(2t+25°)$

21.4 (a) (ⅰ) ∞，90°，(ⅱ) 0.44，450°，(ⅲ) 0.12，26.6°，(ⅳ) 0，0°

(b) (ⅰ) 1，0°，(ⅱ) 0.32，−71.6°，(ⅲ) 0.16，−80.5°，(ⅳ) 0，−90°

21.5 见图 P-12

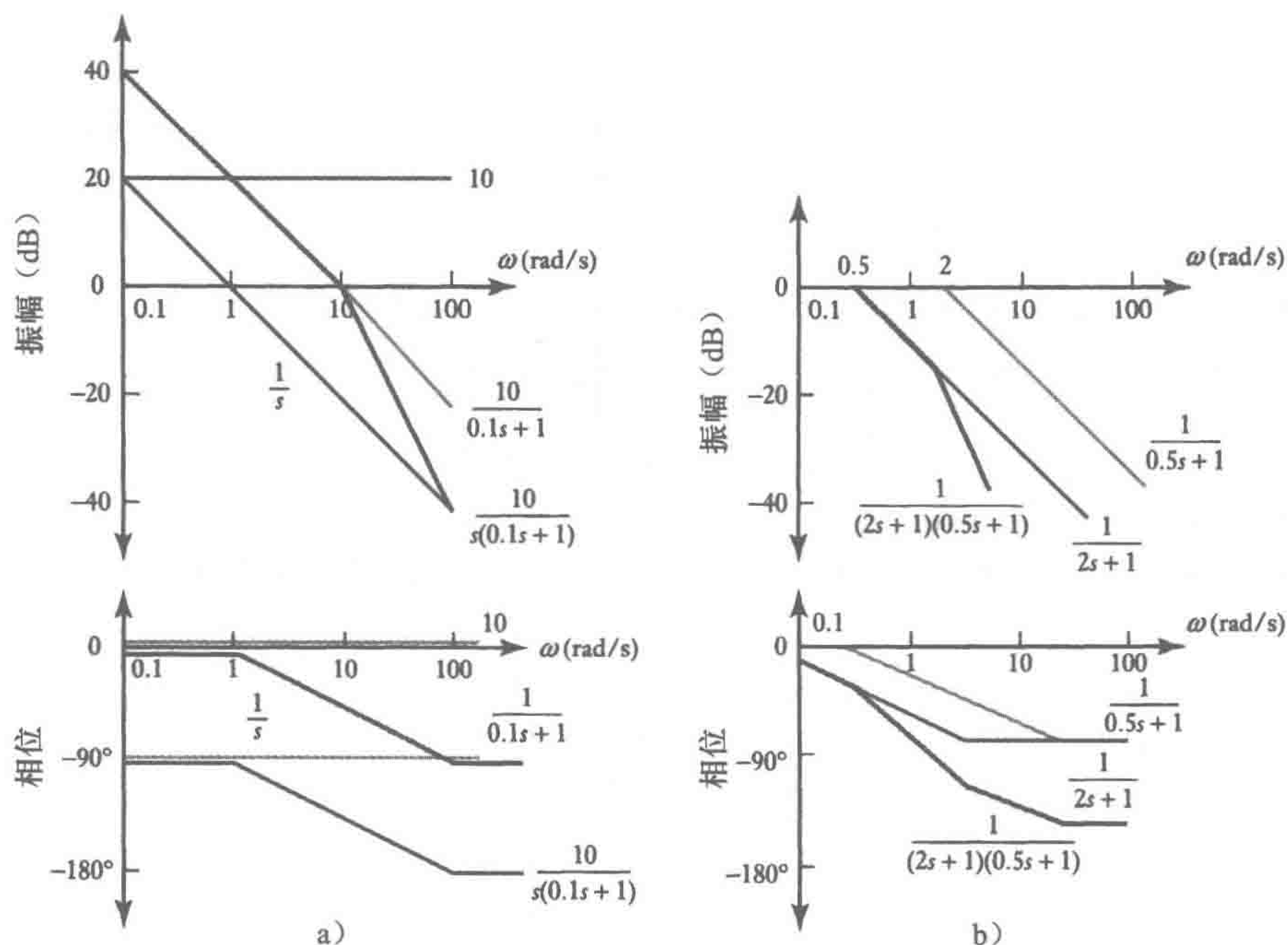


图 P-12　习题 21.5

21.6 (a) $1/s$，(b) $3.2/(1+s)$，(c) $2.0/(s^2+2\zeta s+1)$

(d) $3.2/[(1+s)(0.01s^2+0.2\zeta s+1)]$

第22章

22.1 见22.3节

22.2 (a) 8m, (b) 20m

22.3 (a) 12s, (b) 24s

22.4 5

22.5 见正文。P有偏差，PI和PID无偏差

22.6 3, 666s, 100s

22.7 3, 100s, 25s

22.8 见 (a) 22.12.1节，(b) 22.12.2节，(c) 22.12.3节

22.9 1.6

22.10 时间常量为 c/K_p 的一阶响应

第23章

23.1 例如，可以尝试通过直径和红色的程度来辨别。也可以考虑通过重量。无论什么情况，结果必须能清楚区分硬币的面值。 635

23.2 (a) 1/6, (b) 5/36, (c) 1/10

23.3 0.99

23.4 0.002

23.5 0.625

23.6 0.761

23.7 例如，如果室温小于20℃并且定时器开启，那么热水器也会开启；如果热水器开启，那么泵也会开启；如果泵开启而且室温小于20℃，那么阀就会开启；如果定时器没有开启，那么热水器也不开启；如果室温不小于20℃，那么阀不开启；如果热水器不开启，那么泵不开启。你还可以通过考虑热水器被限制在60℃以下工作这个条件来对此进行改善。

第24章

24.1 可能的解答有：(a) 热电偶，冷端补偿，放大器，ADC，PIA，微处理器，DAC，控制炉子的晶闸管元件，加热元件，(b) 光束传感器，PLC，电磁式输送斜槽导流器，(c) 对每个方向上的运动闭环控制，直流电动机用来驱动笔的运动，微处理器用于比较器和控制器，从光电编码器获得反馈。 636

研究作业

下面是对解答中可能包含的信息做出的简短提示：

24.2 一个典型的ABS系统包含传感器，是电感型的，用来检测每个车轮的转速，通过信号调理把传感器信号转换成5V脉冲。带有程序的微处理器将计算制动时的车轮速度和减速率，当超出设定界限时，微处理器给液压调节器的电磁阀提供输出，从而达到防止制动力增大或者减小制动力的目的。

24.3 打印机是由托架电动机驱动打印头移动来打字的，打完一行后电动机进纸。打印头由电磁驱动引脚组成，通常是一排9个，撞击色带。可以用微控制器来控制输出。更多详细内容参见《Microcontroller Technology：The 68HC11》，P. Spasov（Prentice Hall 1992，1996）。

24.4 CAN总线，信号由以下组成：依次为起始位，表示信息传输目的地和优先级的字，控制位，设置的数据位，CRC位，接收确认位，最后由结束位终止。

设计作业

下面对可行的解决方案做简单提示：

24.5 使用微处理器的数字温度计可能含有：温度传感器（如 LM35）、ADC、ROM（如摩托罗拉 MCM6830 或 Intel 8355）、RAM（如摩托罗拉 MCM6810 或者 Intel 8156）、微处理器（如摩托罗拉 M6800 或 Intel 8085A）以及 LED 显示的驱动器。如果有像摩托罗拉 MC68HC11 或 Intel 8051 这样的微控制器，那么数字温度计中只要有温度传感器、信号调理、微控制器以及 LED 显示驱动器就可以了。

24.6 数字欧姆表可能含有一个单稳多谐振荡器（可以提供宽度为 0.7*RC* 的脉冲）。可以用一系列不同的固定电容来提供不同的电阻范围。采样时间间隔可以由微控制器或者有存储器的微处理器来决定，然后直接通过合适的驱动到 LED 显示。

24.7 可能包括压力传感器（例如半导体传感器摩托罗拉 MPX2100AP），用来将来自传感器的微小微分信号转换成合适电平的信号调理器（例如使用运算放大器的仪表放大器），微控制器（如 MC68HC11），LCD 驱动（如 MC145453），以及四位 LCD 显示器。

24.8 这可以通过使用 M68HC11EVM 板并将 PWM 信号输出到电动机驱动器来解决。其中需要反馈，可以使用光电编码器。

24.9 可以这样设置：对于每个箱子，由电流供给电磁阀操纵气缸推开翼片，使箱子可以沿导槽运动。翼片关闭时箱子留在槽中。在传感器检测到箱子的情况下，指示下一个箱子可以进入导槽。像这样继续一直计数到有 4 个箱子进入了导槽。这时导槽尾端的翼片由另一电磁阀操纵的气缸驱动，使 4 个箱子运到传送带上。安装在导槽尾端的传感器可以显示箱子都被运上了传送带。于是，可使整个过程再次重复。

637

附录 A

A.1 (a) $2/s^2$，(b) $2/(s^2+4)$，(c) e^{-2s}，(d) $sX(s)-2$，(e) $3s^2X(s)$，(f) $1/[s(s+1)]$

A.2 (a) t，(b) $5\cos 3t$，(c) $1+2e^t$，(d) e^{-3t}

A.3 5

附录 B

B.1 255

B.2 (a) 11，(b) 529

B.3 (a) 1A7，(b) 211

B.4 (a) 781，(b) 157

B.5 (a) 10100110，(b) 1101 1101

B.6 (a) 0，(b) 1

B.7 (a) 8，(b) 12

附录 C

C.1 (a) $A\cdot(B+C)$，(b) $(A+B)\cdot(C+D)$，(c) $\bar{A}+B$，(d) $\bar{A}\cdot B$

C.2 (a) $Q=(A\cdot B+C\cdot D)\cdot E$，(b) $Q=(A\cdot B+B)\cdot C$

C.3

A	*B*	*C*	*Q*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1

（续）

A	B	C	Q
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

C.4 (a) $Q = C \cdot (A + D)$，(b) $Q = A \cdot B$，(c) $Q = A \cdot \overline{B} \cdot C + C \cdot D$

C.5 见问题。

C.6 (a) $Q = A + B$，(b) $Q = C + \overline{A} \cdot C$

C.7 (a) $Q = \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{C}$，(b) $Q = A \cdot B \cdot D + A \cdot B \cdot \overline{C} + \overline{C} \cdot D$

C.8 四输入与门，有两个非门，如果正确的组合是 1，1，0，0，则有：
$Q = A \cdot B \cdot \overline{C} \cdot \overline{D}$

638

索　引

索引中的页码为英文原书页码，与书中页边标注的页码一致。

A

C

E

F

I

J

M

N

O

P

Q

R

S

U

V

W

X

Z